DEGREE PROJECT IN MATHEMATICS,
SECOND CYCLE, 30 CREDITS
STOCKHOLM, SWEDEN 2018

Zero-Inflated Hidden Markov Models and Optimal Trading Strategies in High-Frequency Foreign Exchange Trading

JOEL BERHANE

KTH ROYAL INSTITUTE OF TECHNOLOGY
SCHOOL OF ENGINEERING SCIENCES

Zero-Inflated Hidden Markov Models and Optimal Trading Strategies in High-Frequency Foreign Exchange Trading

JOEL BERHANE

Degree Projects in Financial Mathematics (30 ECTS credits)
Degree Programme in Applied and Computational Mathematics
KTH Royal Institute of Technology year 2018
Supervisors at SEB: Pär Hellström
Supervisor at KTH: Jimmy Olsson
Examiner at KTH: Jimmy Olsson

TRITA-SCI-GRU 2018:005
MAT-E 2018:02

Royal Institute of Technology
School of Engineering Sciences
KTH SCI
SE-100 44 Stockholm, Sweden
URL: www.kth.se/sci

Abstract

The properties of high-frequency foreign exchange markets and how well they can be modeled using Hidden Markov Models will be studied in this thesis. Specifically, a Zero-inflated Poisson HMM will be implemented and evaluated for high-frequency price data for the EURSEK exchange rate. Furthermore, a trading strategy aimed at distributing large volumes optimally is developed and evaluated. The results show that the price model performs better than a random walk for some prediction horizons, both when used as a price predictor and as a classifier. The initial tests of the strategy indicate that it has good performance compared to the market benchmark. Both the price model and the strategy needs to undergo more testing before any final conclusions can be made.

Sammanfattning

Egenskaperna hos högfrekventa valutamarknader och hur dessa kan modelleras med Dolda Markovmodeller behandlas i detta examensarbete. Noll-utökade Poisson distributioner, tillsammans med Dolda Markovmodeller, implementeras och utvärderas för högfrekvent växelkursdata för valutaparet EURSEK. Vidare, utvecklas och utvärderas en handelsstrategi med målet att distribuera stora volymer optimalt. Resultaten visar att prismodellen presterar bättre än en slumpvandring för en del prediktionshorisonter, både när den används för prisprediktion och för klassifiering. Initiala tester av strategin indikerar att prestandan är bra jämfört med marknadens prestandamått. Både prismodellen och strategin behöver dock undersökas mer innan några definitiva slutsater kan dras.

Acknowledgements

I would like to thank my supervisor Jimmy Olsson for his invaluable support and guidance through the course of this thesis. Furthermore, I would like to thank my supervisor at SEB, Pär Hellström, for giving me the opportunity to write my thesis at SEB and introducing me to the topic.

Last, but far from least, I want to direct my dearest gratitude and love to my family and friends for all the love and support throughout my studies. Without their endurance, I would not have finished this thesis.

Contents

List of Figures

List of Tables

Nomenclature

Abbreviations

AIC	Akaike Information Criterion
BIC	Bayesian Information Criterion
BW	Baum-Welch
CDF	Cumulative Distribution Function
EM	Expectation-Maximization
FX	Foreign Exchange
GBM	Geometric Brownian Motion
HFT	High-Frequency Trading
HMM	Hidden Markov Model
IC	Information Criterion
PDF	Probability Density Function
TWAP	Time-Weighted Average Price
VWAP	Volume-Weighted Average Price

Notation

$\mathbb{E}$	Expectation of a stochastic variable
$\mathbb{N}$	Non-negative integers
$\mathbb{R}$	Real Numbers
$\mathbb{V}$	Variance of a stochastic variable
A	Transition matrix for a discrete Markov Chain
O_t	Observation at time t
$O_{1:t}$	Observation sequence up until time t

Q_t	Hidden state at time t
$Q_{1:t}$	Hidden state sequence up until time t
D	Number of mixture components of the HMM (including Dirac)
K	Number of states of the HMM

Chapter 1

Introduction

1.1 Hidden Markov Models

Since their inception in the late 1960s, hidden Markov models (HMMs) have found widespread use in various fields and disciplines, with the most prominent examples found in speech recognition and bioinformatics [39]. More recently, HMMs have also found their way into economics and modelling of financial time series [42][14][50]. One of the reasons for their popularity is the extensively developed theory for Markov Chains(MC) and mixture distributions, on which HMMs are based. This is also the case for the estimation procedure, which are based on well-known algorithms, for which the convergence behaviour is known. As such, the theory of HMMs rests on a solid and thoroughly documented theoretical framework.

Another reason for their popularity is their flexibility as descriptive models. Indeed, it is noted in [6] that HMMs can essentially, given sufficient dimension and rich observation distribution, model any distribution. This also explains why HMMs, a by now rather old model, is still popular today even though more sophisticated models and frameworks are available.

1.2 Foreign Exchange

The Foreign Exchange (FX) market is the largest financial market in the world and is an essential cog in the machinery of global economics. Indeed, international trade and globalization would not be possible without the currency markets. It has an average daily turnover close to 5 trillion US dollars [46]. FX is mainly traded by large international companies and banks, including central banks. The market is decentralized and traded on a few large electronic communication networks, compared to stocks which are traded on physical exchanges.

Although most of the world's currencies can be bought and sold, a few currencies overwhelmingly dominate the currency markets. These are the USD, EUR, JPY, GBP and the AUD and together they account for the 160% daily volume (the total volume sums to 200% as purchasing of one currency implies selling of another) [49].

Like all financial markets, technological development has led to an increasing automation of markets. Where currency usually was exchanged trough brokers, now matching algorithms and electronic networks control almost all aspects of the trading. Moreover, algorithmic trading, where computers and models are used to inform and perform trading decisions, is rapidly gaining a larger share of the market [1]. High-frequency trading can be seen as the most extreme iteration of modernization, where trading is essentially completely computerized and trading is performed in micro-seconds, far exceeding the capacity of human traders.

FX is traded on several financial centers around the world 24 hours a day for five days a week. The highest trading activity, and consequently the highest liquidity, is reached when markets have overlapping trading hours. Figure 1.1 below shows operation hours for some of the largest financial centers where FX is traded.

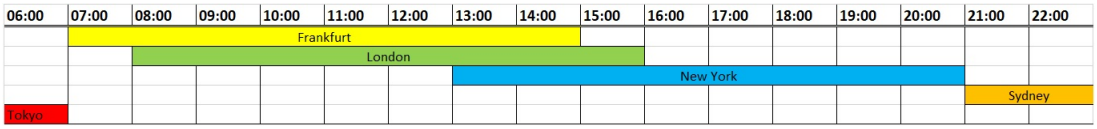

Figure 1.1: *Operation hours for financial centers where FX is heavily traded [46]. Time is expressed in GMT.*

Currencies are quoted using an international system. In EURSEK, for example, EUR is the base currency and SEK is the quote currency. The exchange rate for EURSEK therefore shows the price of 1 EUR in SEK. A trader wanting to sell SEK for EUR buys the currency pair EURSEK and vice versa.

As noted in figure 3.3 below, the trading activity is relatively predictable as a function of the time of the day and it is an important part of trading. Lower market activity not only decreases the rate of change in the bid and ask prices, it also increases the spread between the two, which is an increasing function of the uncertainty in the market. The larger spread can be interpreted as market makers requiring a larger premium for the risk they take when offering to buy and sell. A successful trading strategy should therefore takes into account the time of day in the trading decision.

1.3 HMMs in FX

Much work has been done in macro economics on the modelling of exchange rates, from seminal works on empirical exchange rate models [30] to time series model [4]. The overall consensus is that exchange rates are notoriously difficult to predict based on

economical models of currency. Work has also been done on the modelling of exchange rates in finance with applications to trading. Notable previous works are [25], [43], [48] and [8]. In this literature, both FX and stock markets have been studied, but there is limited research about the HMMs on high-frequency data. In particular, previous research has focused on using HMMs with continuous observation distributions. To the author's knowledge, this is the first public work on discrete HMMs in high-frequency foreign exchange data.

1.4 Thesis objectives

The main objective of the thesis is to formulate, estimate and evaluate a predictive price model for high-frequency foreign exchange data, using Hidden Markov models and zero-inflated Poisson distributions. The second objective is to develop and evaluate a trading strategy for distributing large volumes over times, with the goal of outperforming the market benchmark.

1.5 Outline

The outline of the thesis is as follows. Chapter 2 gives an account of the theory behind the methods and algorithms used in this thesis. It begins by describing different methods from various fields, before moving on to giving an extensive account of the EM-algorithm. The following sections define and describe HMMs by explaining their properties, implications of model assumptions and how to estimate the parameters. The last section in the chapter introduces the main model of study in this thesis, the Zero-Inflated Poisson model.

Chapter 3 describes and displays the data studied in this thesis. Chapter 4 gives a deeper account of FX markets and trading, describing in some detail how markets work and how trading is performed. The second part of the chapter describes the devised strategy and introduces some assumptions made in the modelling.

Chapter 5 describes how the modelling was performed, including details on implementation, made assumptions and fixed parameter values. Some evaluation metrics of the models are defined, together with details on model specifics. Chapter 6 presents the results obtained from the different experiments and shows relevant plots and tables of the calculated values. Chapter 7 gives a thorough discussion of the results, together with the impact of made assumptions and the suitability of the chosen methods for the problem under study. The last chapter, 8, ends with some concluding remarks and suggestion for future research. Figures, tables and derivations excluded in the previous chapters can be found in the appendix, chapter 9.

Chapter 2

Theory

2.1 Preliminaries

2.1.1 Confusion matrix

When evaluating a classifier, or a predictive model where the outputs lie in a discrete set, the performance of the classifier can conveniently be displayed in a confusion matrix. It gives a summary of the results by grouping predictions and their corresponding true values, such that the error in the classifier can be assessed. The rows of the confusion matrix show the distribution of the predictions for each of the true classes. That is, let C and $\hat{C}$ denote the true and the predicted class respectively, both taking values in $\mathscr{C}$. Then the rows show the distribution $P(\hat{C}|C)$ with support on $\mathscr{C}$, and where C is fixed. Conversely, the columns show the distribution $P(C|\hat{C})$. Bayes' theorem can readily be used for converting between the two.

Table 2.1 shows an example of a confusion matrix for a system with 2 classes, 0 and 1. If the entries are counts, then the probabilities above can be calculated as follows

$$\begin{aligned} P(\hat{C}=0|C=0) &= TP/A^*, & P(\hat{C}=1|C=0) &= FN/A^*, \\ P(\hat{C}=0|C=1) &= FP/B^*, & P(\hat{C}=1|C=1) &= TN/B^*. \end{aligned}$$

where the variable names are from the confusion matrix in 2.1.

Two frequently used accuracy measures are the Sensitivity and the Precision, defined as follows

$$\begin{aligned} \text{Sensitivity}(c) &= P(\hat{C}=c|C=c), \\ \text{Precision}(c) &= P(C=c|\hat{C}=c), \end{aligned} \tag{2.1}$$

where $c \in \mathscr{C}$. Sensitivity measures how good the classifier is at detecting (sensing) what the true class is. In the trading setting, this translates to detecting profitable market

		Predicted		
		0	**1**	**total**
True	**0**	**T**rue **P**ositive	**F**alse **N**egative	$A^* = TP + FN$
	1	**F**alse **P**ositive	**T**rue **N**egative	$B^* = TP + FN$
	total	$A = TP + FP$	$B = FN + TN$	

Table 2.1: *Confusion matrix for a classifier with 2 classes. Note that: total* $= A+B = A^*+B^*$.

conditions. Precision measures how accurate the classifier is by looking at the number of correct predictions for each class. Again, this translates to acting on trading indications. These two measures can be used to form a combined accuracy measure known as the F_β-measure, defined as:

$$F_\beta(\mathrm{c}) = (1+\beta^2)\frac{\mathrm{Precision(c)} \cdot \mathrm{Sensitivity(c)}}{\beta^2 \cdot \mathrm{Precision(c)} + \mathrm{Sensitivity(c)}}, \tag{2.2}$$

which is the harmonic mean of the sensitivity and precision, where β is a weighting factor. The F_β-measure can be used for model selection, although it can be biased depending on the problem at hand [38]. In this thesis, it will only be used as a performance measure. Figure 9.3 in the appendix shows a plot of the F-measure and contour curves.

A slightly different definition of precision, compared to the standard one given above, will be used in this thesis. The rationale for this will be explained later on. Let $\mathscr{C} = \{-1, 0, 1\}$. Precision is now defined as follows

$$\begin{aligned} &\mathrm{Precision}(-1) = P(C \neq 1 | \hat{C} = -1), \\ &\mathrm{Precision}(0) = P(C = 0 | \hat{C} = 0), \\ &\mathrm{Precision}(1) = P(C = -1 | \hat{C} = 1). \end{aligned} \tag{2.3}$$

2.1.2 Geometric Brownian Motion

The Geometric Brownian Motion is a widely used mathematical tool for modelling stock prices and it is central in the Black-Scholes model of financial mathematics [16, section 12.3]. Although some of the assumptions of the model are known to be unrealistic or in conflict with empirical observations, it is still widely used due to its properties and

relative simplicity of use. A GBM can be defined using a stochastic differential equation, however, a stochastic process first needs to be defined.

Definition 1 (Stochastic process)**.** *A stochastic process is a collection of random variables* $\{X_i\}_{i\in I}$ *defined on a common probability space* $(\Omega, \mathcal{F}, P)$*, where* Ω *is the sample space,* $\mathcal{F}$ *is a* $\sigma-$*algebra on* Ω*, and* P *is a probability measure defined on* $\mathcal{F}$*.*

With this defintion, a GBM can be defined. A stochastic process S_t is said to follow a GBM if it satisfies the following SDE

$$dS_t = \mu S_t dt + \sigma S_t dB_t, \tag{2.4}$$

where B_t is a Brownian motion, μ is the drift or trend, and σ is the volatility, the last two calculated as percentages. As the names imply, the first term models trend in the price, while the second term is a measure of the variation in the price process.

Maximum likelihood estimates for the parameters μ and σ can readily be derived by noting that an analytic solution of the SDE, using Itō calculus, exists on the following form

$$S_t = S_0 \exp\left(\left(\mu - \frac{\sigma^2}{2}\right)t + \sigma B_t\right). \tag{2.5}$$

The solution S_t is a log-normally distributed variable with mean and variance given as follows

$$\begin{aligned} \mathbb{E}[S_t] &= S_0 e^{\mu t}, \\ \mathbb{V}[S_t] &= S_0^2 e^{2\mu t}\left(e^{\sigma^2 t} - 1\right). \end{aligned} \tag{2.6}$$

2.1.3 Mixture distributions

For a finite set of probability density functions $p_1(x), p_2(x), \ldots, p_m(x)$ and weights $\omega_1, \omega_2, \ldots, \omega_m$, where $\omega_i \geq 0, \forall i$ and $\sum_{i=1}^m \omega_i = 1$, a mixture distribution $f(x)$ can be defined as follows

$$f(x) = \sum_{i=1}^{m} \omega_i p_i(x). \tag{2.7}$$

The component distributions $p_i(x)$ can be either discrete or continuous, with the only difference being that sums are exchanged for integrals. The mixture distribution is itself a proper probability distribution as it is a convex combination of probability distributions. This can be realized by noting that

$$\int_{-\infty}^{\infty} f(x)dx = \int_{-\infty}^{\infty} \sum_{i=1}^{m} \omega_i p_i(x)dx,$$

where the integral is a linear operator and the sum is finite, hence the integral and sum are interchangeable, yielding

$$\begin{aligned}\int_{-\infty}^{\infty} \sum_{i=1}^{m} \omega_i p_i(x) dx &= \sum_{i=1}^{m} \omega_i \int_{-\infty}^{\infty} p_i(x) dx \\ &= \sum_{i=1}^{m} \omega_i = 1. \end{aligned} \tag{2.8}$$

Using the same reasoning, the cumulative distribution function for the mixture distribution can be obtained as follows

$$F(x) = \int_{-\infty}^{x} \sum_{i=1}^{m} \omega_i p_i(s) ds \tag{2.9}$$

$$= \sum_{i=1}^{m} \omega_i \int_{-\infty}^{x} p_i(s) ds \tag{2.10}$$

$$= \sum_{i=1}^{m} \omega_i F_i(x). \tag{2.11}$$

This equation implies that the CDF of a mixture distribution can be obtained from the CDFs of the mixture components.

Using similar reasoning, the expected value of a stochastic variable from the mixture distribution can be derived. Suppose that X is a stochastic variable with PDF $f(x)$ and let $G(\cdot)$ be any function, such that $\mathbb{E}[G(X_i)]$ exists. Then $\mathbb{E}[G(X)]$ can be obtained as follows

$$\begin{aligned}\mathbb{E}[G(X)] &= \int_{-\infty}^{\infty} G(x) f(x) dx \\ &= \int_{-\infty}^{\infty} G(x) \sum_{i=1}^{m} \omega_i p_i(x) dx \\ &= \sum_{i=1}^{m} \omega_i \int_{-\infty}^{\infty} G(x) p_i(x) dx \\ &= \sum_{i=1}^{m} \omega_i \mathbb{E}[G(X_i)], \end{aligned} \tag{2.12}$$

where X_i denotes a stochastic variable with PDF $p_i(x)$. Specifically, the expected value and variance of X is given by

$$\mathbb{E}[X] = \sum_{i=1}^{m} \omega_i \mathbb{E}[X_i], \tag{2.13}$$

$$\mathbb{V}[X] = \sum_{i=1}^{m} \omega_i \Big(\big(\mathbb{E}[X_i] - \mathbb{E}[X] \big)^2 + \mathbb{V}[X_i] \Big). \tag{2.14}$$

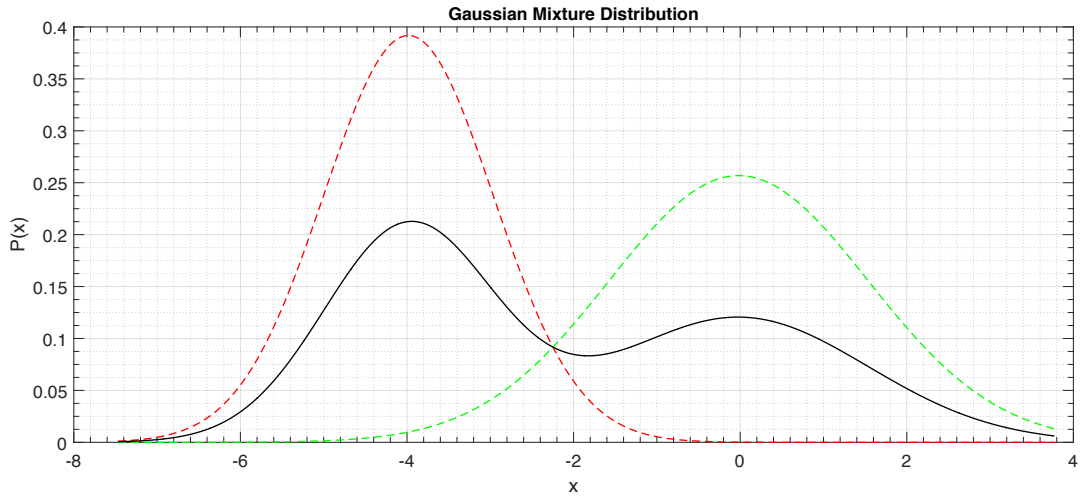

Figure 2.1: *PDF for a mixture of Gaussians. The colored lines represent PDF for univariate Gaussians and the black line is the PDF for resulting mixture distribution with weights* $\omega_1 = \omega_2 = 1/2$.

There is one major difference in mixture modeling when using continuous distributions compared to discrete distributions. It is possible that the likelihood becomes unbounded in the vicinity of some parameter combinations. For example, in a mixture of Gaussians, the likelihood increases without bound when one of the mixture component collapses onto a single point, with mean value equal to the observation and zero variance. The problem arises from the use of densities instead of probabilities. In the discrete case, the likelihood is formed through probabilities, and not densities, ensuring that it is bounded by 1 and 0 [51, p. 11].

2.2 The EM algorithm

The Expectation-Maximization algorithm is one of the most widely used methods in statistics and machine learning for estimating parameters in latent variable models. The algorithm alternates between two steps, the Expectation step and the Maximization step, to find maximum likelihood estimates of the parameters. These estimates may not be global maxima, as the algorithm is only guaranteed to converge to local maxima of the likelihood function.

Made famous in a classic paper by Dempster et al. [10], the different versions of the algorithm had been discovered in previous research (see for example the authors notes

in [10] and [44]). Another classic paper by Wu [47] established convergence results for the algorithm for a larger class of probability distributions than the exponential family.

The EM-algorithm is often used in clustering analysis, where the latent variables indicate which cluster each observation originates from. Another widespread clustering algorithm, which is easier to describe and can be obtained as a simpler case of the EM-algorithm, is the K-means algorithm. It is also often used to generate initial estimates for the EM-algorithm, as it is less computationally intensive. The following sections first describe the K-means algorithm, before moving on to the EM-algorithm and the two steps of the algorithm. The relation between the K-means and EM-algorithm are also explained and an explanation of the convergence properties of the algorithm is given. Much of the material in this section is from [7], unless otherwise stated.

2.2.1 K-means

The K-means algorithm is a method for partitioning a data set of n points into k clusters or groups. Assume that a data set is given, consisting of the points $(x_1, \ldots, x_n)$, where each observation x_i is drawn from a set of K different clusters. Each cluster can represent different distributional properties of the data generating process. Let μ_k denote a prototype observation from cluster k, meaning that each observation from cluster k is similar to μ_k. Given the data set, or the observations $(x_1, \ldots, x_n)$, the objective is to find the correct classification of each observation. That is, the goal is to assign each observation to the cluster that generated it. The problem is that the corresponding cluster for each observation is not observed, i.e. it is latent in the data. The K-means algorithm attempts to solve this problem by finding the best, for which the meaning will be made clear in a moment, assignment of data points to cluster. The idea is that the best assignment is the most likely to have generated the data. This is related to the maximum likelihood principle of parameter estimation.

To further explain the algorithm, it is convenient to introduce some notation. Assume first that the number of clusters k is fixed and that initial estimates for the cluster centers μ_k are given. The data can be multidimensional. Let r_{nk} denote the cluster assignment for the n :th data point, and be defined as follows

$$r_{nk} = \begin{cases} 1, \text{ if } k = \arg\min_j \|x_n - \mu_j\|^2 \\ 0, \text{ otherwise.} \end{cases} \tag{2.15}$$

In other words, each observation is assigned to the closest cluster. Continuing, an objective function can now be defined as follows

$$J(r, \mu) = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|x_n - \mu_k\|^2,$$

which is the sum of the squared Euclidean distances between each observation and its assigned cluster. The goal is now to minimize $J(r, \mu)$, where r denotes all the cluster assignments and μ denotes the cluster means. The minimization can be performed in two steps. Given estimates for all cluster centers μ_k, $J(r, \mu)$ can be maximized with respect to r by simply evaluating r_{nk}, in equation 2.15, for each observation. Once r is found, $J(r, \mu)$ can be minimized with respect to μ by simply setting the derivative to zero, yielding

$$2\sum_{n=1}^{N} r_{nk}(x_n - \mu_k) = 0,$$

which can be solved for μ_k,

$$\mu_k = \frac{\sum_{n=1}^{N} r_{nk} x_n}{\sum_{n=1}^{N} r_{nk}}.$$

The nominator is the sum of all the observations assigned to cluster k and the denominator is the number of observations assigned to cluster k. Hence, μ_k is the mean of cluster k, explaining the name of the algorithm. The two step procedure of minimizing the objective function $J(r, \mu)$ is then continued by alternating between calculating the cluster assignments r_{nk} and the cluster means μ_k. $J(r, \mu)$ is reduced in each step, indicating that the algorithm converges to a minimum value. Note that this minimum is not guaranteed to be the global minimum of $J(r, \mu)$.

Initial estimates for the cluster centers can be obtained by simply randomly sampling k points and setting each of them to be the k^{th} cluster mean. This is not the most efficient way, with respect to the overall convergence of the algorithm, to initialize the algorithm, and various other methods exist for producing initial estimates [17].

2.2.2 Expectation-Maximization

Consider a parametric model where $O_{1:t}$ constitute the observed variables and $Q_{1:t}$ are the corresponding hidden, or latent, variables. Their joint distribution is denoted $P(O_{1:t}, Q_{1:t}|\Theta)$, where Θ denotes a set of parameters. In the following, the sub index $_{1:t}$ will be suppressed to improve readability. All capital letters are to be understood as representing sequences, unless otherwise stated. The goal is now to maximize the following log-likelihood

$$P(O|\Theta) = \sum_{Q} P(O, Q|\Theta), \tag{2.16}$$

where it is assumed that $Q_{1:t}$ is discrete, without loss of generality. Maximization of the log-likelihood is problematic, as it generally becomes very complex for even simple

models. The difficulty arises due to the sum that appears in the log-likelihood function below

$$\log P(O|\Theta) = \log\left(\sum_{Q} P(O, Q|\Theta)\right). \tag{2.17}$$

Suppose now that the hidden variables Q are also observed, so that the complete data consists of $\{O, Q\}$. The log-likelihood function for the complete data now takes the form

$$\log P(O, Q|\Theta),$$

which is generally a much simpler expression to maximize, as the hidden variables generally provide more information about the observations. Thus, an expression for it is desirable. In practice, however, the hidden variables are not observed and knowledge about them is only given through the posterior distribution $P(Q|O, \Theta)$. Hence, the complete-data log-likelihood is not known. The solution is to instead consider the expected value of the complete-data log-likelihood under the poster distribution of the latent variables. Let $\Theta^{'}$ denote a set of fixed parameter values. Assuming that the hidden variables Q are discrete, the log-likelihood of the complete data is given as follows

$$\begin{aligned}\mathcal{Q}(\Theta, \Theta^{'}) =& \mathbb{E}_{\Theta'}\left[\log P(O, Q|\Theta)\middle| O\right] \\ =& \sum_{Q} P(Q|O, \Theta^{'}) \log P(O, Q|\Theta),\end{aligned} \tag{2.18}$$

where $\mathbb{E}_{\Theta'}$ denotes the expectation of the complete data log-likelihood under the posterior distributions. Evaluating this expression is the Expectation-step of the EM-algorithm. This function is often referred to as Baum's auxiliary Q-function, after the seminal work by Baum and his colleagues [5].

After the r.h.s. in equation 2.18 has been evaluated, it is a function of two sets of parameter values, Θ and $\Theta^{'}$. The next step of the algorithm is to maximize $\mathcal{Q}(\Theta, \Theta^{'})$ with respect to the parameter values Θ. That is, the expectation of the complete-data log-likelihood is maximized with respect to the parameters of the joint distribution, which can be written as follows

$$\Theta^{\text{new}} = \arg\max_{\Theta} \mathcal{Q}(\Theta, \Theta^{'}). \tag{2.19}$$

This constitutes the Maximization-step of the EM-algorithm. Once the M-step has been evaluated, the new parameter values are used to re-calculate the posterior distribution of the hidden data. The new parameter values for the posterior distribution are then used to evaluate the $\mathcal{Q}-$function again. In this manner, the EM-algorithm alternates between the E-step and the M-step to produce parameter estimates. The algorithm is

summarized below.

Algorithm 1: The EM-algorithm

Initialization: $\Theta_0, \{O_t\}$

Looping:

for $l = 1, \ldots, l_{max}$ **do**

1. E-step: $\mathcal{Q}(\Theta, \Theta_{l-1}) = \mathbb{E}_{\Theta_{l-1}}\left[\log P(O, Q|\Theta)\middle| O\right]$
2. M-step: $\Theta_l = \arg\max_{\Theta} \mathcal{Q}(\Theta, \Theta_{l-1})$

end

Result: $\{\Theta_l\}_{l=0}^{l_{max}}$

Initial estimates for the EM-algorithm can be obtained by simply randomly sampling parameter values. The algorithm is however known to be sensitive, with respect to both rate of the convergence and exploration of the parameter space, to initial estimates. As mentioned in the section on the K-means algorithm, initial estimates for the EM-algorithm can be obtained by running K-means algorithm for some iterations, which is computationally less intensive.

The relation between the K-means and the EM-algorithm is best understood through the assignment of data points to clusters. The K-means performs hard clustering, where each point is assigned exactly 1 cluster. The EM-algorithm, on the other hand, performs a soft clustering where probabilities-of-belonging are calculated for each point and cluster. That is, in the EM-algorithm the responsibilities from equation 2.15 are replaced with the probabilities $P(Q_t|O_t, \Theta)$, which sums to unity over the possible states Q_t.

2.2.3 Why the EM-algorithm works

The EM-algorithm was explained in the previous section, but no indications were given concerning the convergence of the algorithm. That is the focus of this section. As a first step, note that the complete-data log-likelihood can be rewritten as follows

$$\log P(O, Q|\Theta) = \log P(Q|O, \Theta) + \log P(O|\Theta), \tag{2.20}$$

using the definition of a conditional distribution. In the following expressions, $q(\cdot)$ denotes a probability distribution over the hidden variables. The observed data log-

likelihood can be expanded as follows

$$\begin{aligned}\log P(O|\Theta) =& \log P(O|\Theta) \sum_{Q} q(Q) \\ =& \sum_{Q} q(Q) \Bigg[\log P(O, Q|\Theta) - \log P(Q|O, \Theta) + \log \frac{q(Q)}{q(Q)} \Bigg] \\ =& \sum_{Q} q(Q) \log \frac{P(O, Q|\Theta)}{q(Q)} - \sum_{Q} q(Q) \log \frac{P(Q|O, \Theta)}{q(Q)} \\ =& \mathcal{L}(q, \Theta) + KL(q, P). \end{aligned} \tag{2.21}$$

The second term is the Kullback-Liebler divergence between the two probability distributions $P(\cdot|O, \Theta)$ and $q(\cdot)$ and the first term is a functional over $q(\cdot)$ and a function of Θ. From Gibbs' inequality, it follows that the KL divergence is non-negative, i.e. $KL(q, P) \geq 0$, with equality if and only if $P(\cdot) = q(\cdot)$ almost everywhere. Therefore, the following inequality holds

$$\log P(O|\Theta) \geq \mathcal{L}(q, \Theta). \tag{2.22}$$

The EM-algorithm can now be described through the $\mathcal{L}$ functional. In the E-step, $\mathcal{L}(q, \Theta^{'})$ is maximized with respect to $q(\cdot)$, while holding the known (or old) parameter values $\Theta^{'}$ fixed. By noting that the l.h.s. of equation 2.21, $\log P(O|\Theta)$, does not depend on $q(\cdot)$, and therefore must be constant with respect to $q(\cdot)$, it follows that $\mathcal{L}(q, \Theta^{'})$ is maximized when $KL(q, P) = 0$. Hence, the lower bound for the log-likelihood in equation 2.22 is maximized when $q(\cdot)$ is set to be the posterior distribution of the hidden variables, $P(\cdot|O, \Theta)$.

In the M-step, $\mathcal{L}(q, \Theta)$ is maximized with respect to the parameter values Θ, while $q(\cdot)$ is held fixed. Denote these new parameter values Θ and the old parameter values with $\Theta^{'}$. Unless $\mathcal{L}(q, \Theta)$ is at a maximum, it will increase under the new parameter value and, consequently, so will the log-likelihood, as per equation 2.22. $q(\cdot)$ is determined using the old parameter values, i.e. $q(\cdot) = P(\cdot|O, \Theta^{'})$ almost everywhere. This implies that the KL divergence,

$$KL\Bigg(P(\cdot|O, \Theta^{'}) \Bigg|\Bigg| P(\cdot|O, \Theta) \Bigg),$$

is now non-zero. The total increase in the log-likelihood in equation 2.21 is therefore greater than the increase in the lower bound in equation 2.22.

The importance of the last sentence in the paragraph above can be understood by writing out the function $\mathcal{L}$ as follows

$$\begin{aligned}\mathcal{L}(P(Q|O, \Theta^{'}), \Theta) =& \\ =& \sum_{Q} P(Q|O, \Theta^{'}) \log P(O, Q|\Theta) - \sum_{Q} P(Q|O, \Theta^{'}) \log P(Q|O, \Theta^{'}) \\ =& \mathcal{Q}(\Theta, \Theta^{'}) + H(\Theta^{'}), \end{aligned} \tag{2.23}$$

where the first term is the complete-data log-likelihood from equation 2.18 and $H(\Theta^{'})$ is the negative entropy. The second term is a constant with respect to Θ. Hence, maximizing $\mathcal{L}(q, \Theta)$ in the M-step is actually maximizing the complete-data log-likelihood.

As a last step, the EM-algorithm can be shown to be a non-decreasing iterative algorithm. Let Θ and $\Theta^{'}$ denote new and old parameter values, respectively, and let P_{Θ} denote $P(Q|O, \Theta)$. It then follows that

$$\log P(O|\Theta) - \log P(O|\Theta^{'}) = \\ = \Big(\mathcal{L}(q, \Theta) - \mathcal{L}(q, \Theta^{'})\Big) + \Big(KL(q||P_{\Theta}) - KL(q||P_{\Theta'})\Big).$$

In the E-step, $q(\cdot)$ was set to be equal to $P_{\Theta'}$ almost everywhere. This implies that the first and second term in the second bracket are non-negative and zero, respectively. In the M-step, $\mathcal{L}(q, \Theta)$ was maximized with respect to Θ. Hence, the first bracket is non-negative from which it follows that

$$\log P(O|\Theta) \geq \log P(O|\Theta^{'}),$$

with equality if and only if the log-likelihood is at a maximum. This demonstrates that the log-likelihood is non-decreasing in the EM-algorithm.

2.2.4 Extensions of the EM-algorithm

The EM-algorithm as presented here is the standard version, which is useful when all quantities involved can be written down explicitly. This is the case when the state-space for the underlying Markov chain is finite. When this is not the case, the E-step of the algorithm becomes intractable. Sequential Monte Carlo methods are a large class of methods for solving filtering problems when the EM-algorithms can not be used.

It is also possible that the derivative in the M-step yields complex or intractable expression. Several extensions of the EM-algorithm exists in this case where different methods are used to somehow maximize the $\mathcal{Q}-$function with respect to some of the parameters.

2.3 Hidden Markov models

In this section, HMMs are formally defined and their properties are explained. The section begins with a description of Markov Chains, as they are essential to the theory of HMMs.

The material in the sections below is derived from many different sources. The exposition mainly follows that in [39] and [6].

2.3.1 Markov Chains

Definition 2 (Markov Chain)**.** *Let $\{Q_t\}_{t\in T}$ be a discrete-valued stochastic process. The stochastic process is said to be a Markov process if it satisfies the following Markov property*

$$P(Q_{t+1}|Q_{1:t}) = P(Q_{t+1}|Q_t). \tag{2.24}$$

A Markov chain is a discrete-valued stochastic process satisfying the Markov property.

In words, for a Markov chain, the future only depends on the past through the present. A MC is said to be time-homogeneous if the following holds true

$$P(Q_{t+h+1}|Q_{t+h}) = P(Q_{t+1}|Q_t), \tag{2.25}$$

for any h, meaning that the distribution of the MC does not change with time. A time-homogeneous finite state MC, where Q_t can only take values in a finite set K, can be characterized by a transition matrix, which is a square matrix with dimension given by the size K. For example, the a_{ij}^{th} element in a transition matrix, where i denotes the row and j denotes the column, is given by the transition probability $P(Q_{t+1} = j|Q_t = i)$, where $i, j \in K$. Transition probabilities for multiple steps can easily be obtained using the following result from Chapman and Kolmogorov [12, p. 9].

Theorem 1 (Chapman-Kolmogorov)**.** *Let $A^{(t)}$ denote the $t-$step transition matrix , i.e. the matrix where the elements give probabilities of the following form $P(Q_t = j|Q_0 = i)$. Then the following equality holds*

$$A^{(t+s)} = A^{(t)}A^{(s)}. \tag{2.26}$$

From the Chapman-Kolmogorov equations, it also follows that $A^{(t)} = A^t$ where the r.h.s. denotes the t^{th} power of the matrix A.

Several important properties of the MC can be explained in terms of the transition matrix A. A MC is said to be irreducible if, loosely stated, it is possible to reach every state from any state. The meaning of irreducible can be defined formally using set theory but, here, it is enough to note that a MC with a transition matrix where all elements are positive is irreducible.

Each state in set K for the MC has a period, which is defined as follows for any state $i \in K$

$$k = \gcd\{n > 0 : P(Q_t = i|Q_0 = i) > 0\}.$$

If $k = 1$ for all states in K, the MC is said to be aperiodic. Hence, a MC with a transition matrix where all elements are positive, is aperiodic.

There is a special type of distribution for MCs, called a stationary distribution and it is defined as follows.

Definition 3 (Stationary distribution)**.** *Let A be the transition matrix of a finite-state, time-homogeneous irreducible MC with dimension K. A distribution π is said to be a stationary distribution if it satisfies the following conditions:*

$$\begin{aligned} 0 \leq \pi_i \leq 1, \\ \sum_{i \in K} \pi_i = 1, \\ \pi A = \pi. \end{aligned} \tag{2.27}$$

The stationary distribution is a probability distribution over the state-space of the MC and from the last equation, it is clear that it corresponds to the eigenvalue 1 of the transition matrix, with the eigenvector π. This is the Perron-Frobenius theorem applied to stochastic matrices [31]. The theorem also states that all other eigenvalues of the transition matrix are smaller than 1. When A is finite and irreducible, then the stationary distribution π is unique [12, p. 22]. When the stationary distribution exists and is unique, the following convergence theorem holds true for the MC.

Theorem 2 (Convergence theorem)**.** *[12, p. 26] Let $\{Q_t\}_{t \in T}$ denote a finite-state, time-homogeneous irreducible MC, with transition matrix A and state-space K. If this MC is aperiodic and there exists a stationary distribution π, then, $\forall i \in K$*

$$\lim_{t \to \infty} a_{ij}^{(t)} = \pi_j, \tag{2.28}$$

where $a_{ij}^{(t)}$ denotes the following transition probability $P(Q_t = j | Q_0 = i)$.

In other words, this theorem states that the long-run probability of the MC being in a state j is given by the probability of the state, π_j, in stationary distribution π. The stationary distribution can be found by solving equation 2.27, together with the unity sum constraint.

For diagonalizable transition matrices, A can be decomposed into the form $A = VDV^{-1}$, where D is a diagonal matrix containing all the eigenvalues of A and V is a matrix containing the corresponding eigenvectors as column matrices. The convergence of the transition matrix can then be characterized using the eigenvalues as follows

$$\begin{aligned} A^t =& (VDV^{-1})^t \\ =& VDV^{-1}VDV^{-1} \cdot \ldots \cdot VDV^{-1} \\ =& VD^tV^{-1}. \end{aligned} \tag{2.29}$$

Since D is a diagonal matrix, D^t can be calculated simply by taking the t :th power of the eigenvalues. Returning to the Perron-Frobenius theorem above, it then follows that,

using the convergence theorem,

$$\lim_{t\to\infty} A^t = \lim_{t\to\infty} VD^tV^{-1} \tag{2.30}$$

$$= \lim_{t\to\infty} V \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & \lambda_1^t & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & 0 & \lambda_n^t \end{bmatrix} V^{-1} \tag{2.31}$$

$$= V \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & 0 & 0 \end{bmatrix} V^{-1} \tag{2.32}$$

$$= \begin{bmatrix} \pi \\ \pi \\ \vdots \\ \pi \end{bmatrix}. \tag{2.33}$$

The error made when approximating A^t with the stationary distribution is then determined by the largest eigenvalue of the transition matrix.

2.3.2 HMMs

In HMMs where the underlying MC is discrete, a Hidden Markov Model can be defined in terms of the conditional independence properties of the model. In the following, the $\perp\!\!\!\perp$ symbols denote independence in the probabilistic sense and $\mid$ denotes a conditional probability.

Definition 4 (Hidden Markov Model)**.** *A Hidden Markov Model is a collection of random variables $\{(Q_t, O_t)\}_{t\geq 0}^T$, where $\{Q_t\}_{t\geq 0}^T$ is a discrete stochastic process forming a Markov Chain and $\{O_t\}_{t\geq 0}^T$ is a general discrete time stochastic process that can be either discrete or continuous, satisfying the following conditional independence properties*

$$\begin{aligned} \{Q_{t:T}, O_{t:T}\} &\perp\!\!\!\perp \{Q_{1:t-2}, O_{1:t-1}\} \mid Q_{t-1}, \\ O_t &\perp\!\!\!\perp \{Q_{\neg t}, O_{\neg t}\} \mid Q_t, \end{aligned} \tag{2.34}$$

for all $t = 1, \dots, T$.

Several conditional independence properties are induced from the two equations above. The first one states that the future and the past are conditionally independent, given the present. This in turn implies that $Q_t \perp\!\!\!\perp Q_{1:t-2} \mid Q_{t-1}$, implying that $\{Q_t\}_{t\geq 0}^T$ form a discrete MC, which actually is not necessary to include in the definition of the HMM. The second equation states that the observations $\{O_t\}_{t\geq 0}^T$ are conditionally independent, given the corresponding states.

The conditional independence properties of a HMM suggest that the joint probability over the hidden and observed variables (together, these are called the complete data) can be factorized.

$$\begin{aligned} P(O_{1:T}, Q_{1:T}) =& P(O_T, Q_T \mid O_{1:T-1}, Q_{1:T-1}) P(O_{1:T-1}, Q_{1:T-1}) \\ =& P(O_T \mid Q_T, O_{1:T-1}, Q_{1:T-1}) P(Q_T \mid O_{1:T-1}, Q_{1:T-1}) \\ & \times P(O_{1:T-1}, Q_{1:T-1}), \end{aligned} \tag{2.35}$$

where the equality signs follow from the definition of a conditional distribution. From the conditional independence properties of the HMM, it then follows that

$$P(O_{1:T}, Q_{1:T}) = P(O_T \mid Q_t) P(Q_T \mid Q_{T-1}) P(O_{1:T-1}, Q_{1:T-1}). \tag{2.36}$$

The first factor follows from the second property in equation (2.34), while the second equality is the Markov property of the hidden variables. Repeating this procedure for the last factor in the product, $P(O_{1:T-1}, Q_{1:T-1})$, yields the following factorization of the joint distribution

$$P(O_{1:T}, Q_{1:T}) = P(Q_1) \prod_{t=2}^{T} P(Q_t \mid Q_{t-1}) \prod_{t=1}^{T} P(O_t \mid Q_t). \tag{2.37}$$

This factorization is convenient, as it demonstrates the constituents of the HMM. The first factor represent the initial distribution over the hidden states. The second factor represent the transition probabilities of the underlying MC and the last factor represents the observation (or emission) distributions. Together, these distributions determine the HMM.

Some intuition about the flexibility when using latent variables can be gained through an example of a HMM. Suppose that the returns of a financial asset follow a t-distribution. It is plausible that financial markets can display different behaviour at different times. Specifically, it is possible that the market has different states where the market participants, and consequently the returns, shows similar trends and volatilities. For example, the market might have an "up-state" with overwhelmingly positive returns, a "stagnant-state" where the returns do not appear to show any trend and have large variation and a "down-state", where the returns are mainly negative. Each of these states could appear in an arbitrary order and exist for different periods of time. While a mixture distribution could be used to model the total output of the data, it can not model the temporal properties of the data, which form a time series.

In a HMM, the different states of the market could be represented by different states of the hidden variables. The transition matrix of the underlying HMM would then capture how the market switches between states. The behaviour of the returns is then described by the parameters of the observation distributions in each states. That is, the different values for the parameters of the observation distributions allow for different means and variances of the returns in each state.

Predictive distribution

In order to generate predictions from the HMM, the predictive distribution must first be derived. For an observation sequence $O_{1:t}$ and a time $s \geq 1$, the predictive distribution $P(O_{t+s}|O_{1:t})$ can be derived as follow,

$$\begin{aligned} P(O_{t+s}|O_{1:t}) &= \sum_{Q_{t+s}} \sum_{Q_t} P(O_{t+s}, Q_{t+s}, Q_t|O_{1:t}) \\ &= \sum_{Q_{t+s}} \sum_{Q_t} P(O_{t+s}|Q_{t+s}, Q_t, O_{1:t}) P(Q_{t+s}, Q_t|O_{1:t}) \\ &= \sum_{Q_{t+s}} \sum_{Q_t} P(O_{t+s}|Q_{t+s}) P(Q_{t+s}|Q_t, O_{1:t}) P(Q_t|O_{1:t}) \\ &= \sum_{Q_{t+s}} P(O_{t+s}|Q_{t+s}) \sum_{Q_t} P(Q_{t+s}|Q_t) P(Q_t|O_{1:t}). \end{aligned} \tag{2.38}$$

The second equality is simply the definition of a conditional distribution. The third equality follows from the conditional independence property of HMMs. The final expression is obtained by using the Markov property of the MC and collecting terms. The first term in this expression is the emission density for the state Q_{t+s}. The first term in the second sum is the probability of moving from state Q_t to Q_{t+s} in s steps in the underlying MC.

The expression in equation 2.38 can be simplified by defining the following function

$$V(Q_{t+s}) \triangleq \sum_{Q_t} P(Q_{t+s}|Q_t) P(Q_t|O_{1:t}). \tag{2.39}$$

Using this in equation 2.38, yields the following expression for the predictive distribution

$$P(O_{t+s}|O_{1:t}) = \sum_{Q_{t+s}} P(O_{t+s}|Q_{t+s}) \cdot V(Q_{t+s}). \tag{2.40}$$

In this form, it's evident that the predictive distribution is a mixture distribution, with weights $V(Q_{t+s})$ and mixture components $P(O_{t+s}|Q_{t+s})$, which are mixture distributions themselves. The sampling scheme becomes identical to that of mixture distributions, with the addition of a second level due to the emission distributions also being mixtures. To verify that 2.38 (or 2.40) is a proper probability distribution, note that

$$\begin{aligned} \sum_{Q_{t+s}} V(Q_{t+s}) &= \sum_{Q_{t+s}} \sum_{Q_t} P(Q_{t+s}|Q_t) P(Q_t|O_{1:t}) \\ &= \sum_{Q_t} P(Q_t|O_{1:t}) \sum_{Q_{t+s}} P(Q_{t+s}|Q_t) \\ &= 1. \end{aligned}$$

A few points are worth to emphasize regarding the $V(Q_{t+s})$. For the first term in 2.39, using the Chapman-Kolmogorov equation, it follows that $P(Q_{t+s}|Q_t)$ is obtained by taking the s^{th} power of the transition matrix and choosing the appropriate element in the resulting matrix. The second factor is defined in equation 2.47 and it can be rewritten as follows

$$P(Q_t|O_{1:t}) = \frac{P(Q_t, O_{1:t})}{P(O_{1:1})} = \frac{\alpha_t(Q_t)}{\sum_{Q_t} \alpha_t(Q_t)}. \tag{2.41}$$

Hence, $V(Q_{t+s})$, can be expressed as follows

$$V(Q_{t+s}) = \sum_{Q_t} A^s_{Q_t,Q_{t+s}} \frac{\alpha_t(Q_t)}{\sum\limits_{Q_r} \alpha_t(Q_r)}. \tag{2.42}$$

If the underlying MC has a stationary distribution δ, then the transition matrix will converge to δ as s becomes large. This yields a slight simplification of $V(Q_{t+s})$ as follows: let s be larger than some threshold value n. Then,

$$\begin{aligned} V(Q_{t+s}) =& \sum_{Q_t} A^s_{Q_t,Q_{t+s}} \frac{\alpha_t(Q_t)}{\sum\limits_{Q_r} \alpha_t(Q_r)} \\ \approx& \sum_{Q_t} \delta(Q_{t+s}) \frac{\alpha_t(Q_t)}{\sum\limits_{Q_r} \alpha_t(Q_r)} \\ =& \delta(Q_{t+s}). \end{aligned}$$

Consequently, when the hidden MC has converged to its stationary distribution, the predictive distribution is identical for all future prediction horizons and the dependence of the posterior distribution on the current hidden state is lost.

3 Fundamental problems for HMMs

The disassembled joint probability distribution in equation 2.37 highlights the different parts of a HMM necessary for applications. The first factor is the initial distribution, usually denoted by π, of the HMM over the possible states for the hidden distribution such that

$$0 \leq \pi_i \leq 1, \quad \sum_{i=1}^{K} \pi_i = 1,$$

where K is the number of states, or equivalently, the dimension of the HMM. The second factor represents the transitions of the MC and is determined by the elements of transition matrix, denoted by A. The last factor represents the observation distributions of

the observed variables, denoted by B. These are usually chosen to be distributions from parametric families or mixture distributions, in which case they are indexed by parameters. Together with K and D (the number of mixture components), these factors make up the HMM and are denoted by $\Theta \triangleq (\pi, A, B)$. In reverse, these are the parameters that are required for a complete specification of the HMM.

Given the specification of the HMM described above, some questions naturally arise. The 3 main problems for HMMs, as presented in [39], are as follows:

1. Given an observation sequence $O_{1:T}$ and a model $\Theta = (\pi, A, B)$, what is the likelihood of the observation sequence under Θ, i.e. $P(O|\Theta) =?$
2. Given an observation sequence $O_{1:T}$ and a model $\Theta = (\pi, A, B)$, how is the corresponding hidden sequence found?
3. Given an observation sequence $O_{1:T}$, how are the parameters in Θ adjusted to maximize $P(O_{1:T}|\Theta)$?

These 3 questions will be addressed in the following sections, in the same order as above.

2.3.3 The Forward-Backward algorithm

The FB algorithm is a method to efficiently evaluate the likelihood of a HMM, utilizing the conditional independence properties of the model. Simply evaluating the likelihood of an observation sequence by enumerating all possible state sequences requires on the order of K^T calculations, which quickly becomes intractable for even small sequences as $K = 5, T = 100$ yields $5^{100} \approx 10^{72}$.

The operation of the Forward algorithm can be demonstrated with the help of 2 quantities. First, the joint distribution $P(O_{1:t}, Q_t, Q_{t-1})$ can be expanded as follows

$$\begin{aligned} P(O_{1:t}, Q_t, Q_{t-1}) =& P(O_{1:t-1}, O_t, Q_t, Q_{t-1}) \\ =& P(O_t, Q_t|O_{1:t-1}, Q_{t-1})P(O_{1:t-1}, Q_{t-1}) \\ =& P(O_t|Q_t, O_{1:t-1}, Q_{t-1})P(Q_t|O_{1:t-1}, Q_{t-1})P(O_{1:t-1}, Q_{t-1}) \\ =& P(O_t|Q_t)P(Q_t|Q_{t-1})P(O_{1:t-1}, Q_{t-1}). \end{aligned}$$

The second and third equality follow from the definition of a conditional distribution. The last equality follows from the conditional independence property of the HMM and the Markov property for the unobserved state process. Second, the last factor in equation 2.43 can be decomposed as follows

$$\begin{aligned} P(O_{1:t}, Q_t) &= \sum_{Q_{t-1}} P(O_{1:t}, Q_t, Q_{t-1}) \\ &= \sum_{Q_{t-1}} P(O_t|Q_t)P(Q_t|Q_{t-1})P(O_{1:t-1}, Q_{t-1}), \end{aligned} \tag{2.43}$$

where the second equality follows from equation 2.43. This result suggest the following recursion, introducing the α variable as $\alpha_{Q_t}(t) \triangleq P(O_{1:t}, Q_t)$,

$$\alpha_{Q_t}(t) = \Big[\sum_{Q_{t-1}} P(Q_t|Q_{t-1}) \alpha_{Q_{t-1}}(t-1) \Big] P(O_t|Q_t). \tag{2.44}$$

This is the forward recursion, summarized in algorithm 2. The likelihood can now easily be obtained by summing the $\alpha_{Q_T}(T)$ variable over the hidden states, i.e.

$$P(O_{1:t}) = \sum_{Q_t} \alpha_{Q_t}(t).$$

This calculation is much more efficient than simply enumerating all possible states, utilizing the finer structure of the HMM, and requires on the order of K^2T calculations.

Algorithm 2: The Forward algorithm

Initialization:
$\alpha_1(i) = \pi_i b_i(O_1), \quad 1 \leq i \leq K$

Recursion:
for $t = 1, \ldots, T-1$ **do**
  for $j = 1, \ldots, K$ **do**
    $\alpha_{t+1}(j) = \Big[\sum_{i=1}^{K} \alpha_t(i) a_{ij} \Big] b_j(O_{t+1})$
  end
end

Result: $P(O_{1:T}) = \sum_{i=1}^{N} \alpha_T(i)$

Similarly, a backward recursion can be derived. The distribution $P(O_{t+1:T}|Q_t)$ can be decomposed as follows

$$\begin{aligned} P(O_{t+1:T}|Q_t) &= \sum_{Q_{t+1}} P(O_{t+2:T}, O_{t+1}, Q_{t+1}|Q_t) \\ &= \sum_{Q_{t+1}} P(O_{t+2:T}|O_{t+1}, Q_{t+1}, Q_t) P(O_{t+1}|Q_{t+1}, Q_t) P(Q_{t+1}|Q_t) \\ &= \sum_{Q_{t+1}} P(O_{t+2:T}|Q_{t+1}) P(O_{t+1}|Q_{t+1}) P(Q_{t+1}|Q_t). \end{aligned} \tag{2.45}$$

The first and second equality follow from the definition of a conditional distribution. The third equality follows from the first conditional independence property of the HMM,

stated in equation 2.34. Defining the β variable as $\beta_{Q_t}(t) = P(O_{t+1:T}|Q_t)$, equation 2.45 suggest the following recursion:

$$\beta_{Q_t}(t) = \sum_{Q_{t+1}} \beta_{Q_{t+1}}(t+1)P(O_{t+1}|Q_{t+1})P(Q_{t+1}|Q_t). \tag{2.46}$$

The likelihood of an observation sequence can again be obtained by properly summing over the β variable. The Backward algorithm is summarized in algorithm 3 below.

Algorithm 3: The Backward algorithm

Initialization:
$\beta_T(i) = 1, \quad 1 \leq i \leq K$

Recursion:
for *t=T-1,...,1,* $1 \leq j \leq K$ **do**
$\beta_t(i) = \sum_{j=1}^{K} a_{ij} b_j(O_{t+1})\beta_{t+1}(j)$
end

Result: $\beta_t(i) = P(O_{t+1:T}|q_T = S_i|\Theta)$

Each of the two algorithms can be used separately to calculate the likelihood of a model. They are, however, both necessary when estimating the parameters of a HMM using the Baum-Welch algorithm, which is the EM-algorithm for HMMs.

2.3.4 The Viterbi algorithm

While the objective is clear when calculating the likelihood, finding the unobserved state sequence, responsible for generating the data, is more diffuse. Specifically, an observation sequence can be generated from many different state sequences for the underlying MC. In order to select one of these sequences, an optimality criterion is required. A widely used criterion is to find the state sequence that maximizes the posterior distribution of the hidden states $P(Q_{1:t}|O_{1:t}, \Theta)$. This objective is equivalent to maximizing $P(Q_{1:t}, O_{1:t}|\Theta) = P(Q_{1:t}|O_{1:t}, \Theta)P(O_{1:t}, \Theta)$ with respect to the sequence $Q_{1:t}$. An algorithm exists, based on dynamic programming methods, for finding the sequence of hidden states that maximizes $P(Q_{1:t}, O_{1:t}|\Theta)$, called the Viterbi algorithm. It can be explained by first defining the following quantity

$$\delta_t(i) = \max_{Q_1,\ldots,Q_{t-1}} P(Q_{1:t-1}, O_{1:t-1}, Q_t = i|\Theta),$$

which is the single path $Q_{1:t-1}$ with the highest probability, given the observation and the parameters, up to time $t-1$ and ending on state i at time t. The theory of dynamic

programming then suggest the following induction

$$\delta_{t+1}(j) = \max_i \left[\delta_t(i)_{a_{ij}}\right] b_j(O_{t+1}).$$

Note that $\delta_t(j)$ is calculated for each time point and hidden state j, i.e. for each state at any time, and stored. The optimal sequence is then retrieved by finding the state that maximizes $\delta_T(i)$, where T is the last time point, and then backtracking the state sequence to find the optimal path. The Viterbi algorithm is summarized in algorithm 4 below.

One important feature of the Viterbi algorithm is that it includes the state transition in the calculations, meaning that impossible paths (where some transition has probability $a_{ij} = 0$) are excluded.

Algorithm 4: The Viterbi algorithm

Initialization:

$\psi_1(i) = 0,$

$\delta_1(i) = \pi_i b_i(O_1),\ i = 1, \ldots, K$

Recursion:

for $t = 2, \ldots, T$ **do**

  for $j = 1, \ldots, K$ **do**

    $\delta_t(j) = \max_{1 \le i \le K} \left[\delta_{t-1}(i) a_{ij}\right] b_j(O_t)$

    $\psi_t(j) = \arg\max_{1 \le i \le K} \left[\delta_{t-1}(i) a_{ij}\right]$

  end

end

Termination:

$\max_Q P(O, Q|\Theta) = \max_{1 \le i \le K} \left[\delta_T(i)\right]$

$Q_T^* = \arg\max_{1 \le i \le K} \left[\delta_T(i)\right]$

Backtracking:

for $t = T - 1, \ldots, 1$ **do**

  $Q_t^* = \psi_{t+1}(Q_{t+1}^*)$

end

Result: $\{Q_{1:T}^*\}$

2.3.5 The Baum-Welch algorithm

Originally developed in the 1960s, together with the formulation of HMMs, the Baum-Welch algorithm is a collection of algorithms for estimating the parameters of a HMM. Specifically, it iterates between using the Forward and Backward algorithms to obtain estimates for the posterior distribution of the hidden states, and then uses these estimates in the EM-algorithm to obtain updates for the parameters of the hidden MC and the observation distributions.

Two new variables are required in the calculations. Define

$$\gamma_t(i) = P(Q_t|O_{1:t}, \Theta), \tag{2.47}$$

i.e. the probability of being in state Q_t at time t, and

$$\varepsilon_t(i,j) = P(Q_t, Q_{t+1}|O_{1:t}, \Theta), \tag{2.48}$$

i.e. the probability of being in state Q_t at time t and $Q_t + 1$ at time $t + 1$. The main algorithms in the BW-algorithm have already been described in earlier sections. As such, the BW-algorithm is not described further here and instead summarized below in algorithm 5. A full derivation of the estimation equations for the HMM, including the equations given in algorithm 5, can be found in the appendix.

Note that the set of equations given in algorithm 5 are identical for all mixture distributions and independent of the form of the observation distributions. These parameters, i.e. the subset of parameters of the HMM Θ not stated above, are also updated in the M-step. Equations for the remaining parameters can be found in the appendix.

The BW-algorithm is essentially the EM-algorithm for HMMs and the names will be used interchangeably when discussing parameter estimation for the HMM in the remaining sections.

Algorithm 5: The Baum-Welch algorithm

Initialization: Θ_0, $\{O_{1:T}\}$

Looping:

for $l = 1, \ldots, l_{max}$ **do**

1. Forward-Backward calculations:

$$\alpha_1(i) = \pi_i b_i(O_1), \;\; \beta_T(i) = 1$$

$$\alpha_t(i) = \left[\sum_{j=1}^{K} \alpha_{t-1}(j) a_{ji}\right] b_j(O_t), \;\; \beta_t(i) = \sum_{j=1}^{K} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)$$

$$\text{for } 1 \leq i \leq K,\, 1 \leq t \leq T-1$$

2. E-step:

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^{K} \alpha_t(j)\beta_t(j)}, \;\; \xi_t(i,j) = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum\limits_{i=1}^{N}\sum\limits_{j=1}^{N} \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_t(j)}$$

$$\text{for } 1 \leq i \leq K,\, 1 \leq j \leq K,\; 1 \leq t \leq T-1$$

3. M-step:

$$\pi_i = \frac{\gamma_1(i)}{\sum\limits_{j=1}^{K} \gamma_1(j)}, \quad a_{ij} = \frac{\sum\limits_{t=1}^{T} \varepsilon_t(i,j)}{\sum\limits_{k=1}^{K}\sum\limits_{t=1}^{T} \varepsilon_t(i,k)}$$

$$w_{kd} = \frac{\sum\limits_{t=1}^{T} \gamma_t(k,d)}{\sum\limits_{t=1}^{T}\sum\limits_{r=1}^{D} \gamma_t(k,r)}$$

$$\text{for } 1 \leq i \leq K,\, 1 \leq j \leq K,\; 1 \leq k \leq K,\; 1 \leq d \leq D$$

end

Result: $\{\Theta_l\}_{l=0}^{l_{max}}$

2.4 The Zero-Inflated Poisson

The Zero-inflated Poisson(ZIP) is a special case of the more general Zero-inflated models, which are probabilistic models where the probability of observing a zero is inflated in some way. The ZIP is the most famous in this class of models, originally devised in the study of manufacturing quality [23]. Parameters of the ZIP models were traditionally estimated using different forms of regression. Later, ZIP models were used in HMMs in different fields where data generally represents counts [36][45][11].

In this thesis, ZIP mixture models used are on the following form

$$P(O = o) = \mathbb{I}(o)_{[0]} \times w_0 + \sum_{d=1}^{D} \frac{\lambda_d^{o_j} e^{-\lambda_d}}{o_j!} \times w_d \tag{2.49}$$

where w_d are weights for each component, summing to unity. In words, the ZIP models are mixtures of a Dirac component at zero and D Poisson components. The inflation of zeros can be demonstrated by noting that

$$P(O) = \begin{cases} w_0 + \sum\limits_{d=1}^{D} e^{-\lambda_d} w_d, & O = 0 \\ \sum\limits_{d=1}^{D} \frac{\lambda_d^O e^{-\lambda_d}}{O!} w_d, & O \neq 0 \end{cases}$$

The probability of observing a zero is therefore inflated by the w_0 weight component of the mixture distributions.

These mixture distributions will be used as emission, or observation, distributions for the hidden states of the HMMs defined in the previous section. As a short-hand notation, let ZIP(K,D) an HMM with emission distributions given by equation 2.49, with D mixture components ($D-1$ Poissons) and K states. The full derivation of the quantities and parameter estimation equations required in the BW-algorithm can be found in the appendix.

Chapter 3

Data

This section describes and displays all the data studied in the thesis.

3.1 Market data

The price data obtained from SEB and consists of exchange rates, bids and asks, for the currency pairs EURSEK and EURUSD, recorded from 00:00 to 22:00 on the 12^{th} of January 2017. Specifically, the data consist of the limit order book for the stated period, which contains

- All the bid prices (the prices traders are willing to buy at)
- The corresponding volumes
- All the ask prices (the prices traders are willing to sell at)
- The corresponding volumes

The depth of the order book is 5 levels, that is the 5 best bids and asks are shown. A snapshot of the order book is shown in figure 3.2.

The exchange rate gives the cost of one unit of currency expressed in the other currency, with the convention that the EURSEK is the price of 1 EUR in SEK. The data is recorded on a tick-by-tick basis, which means that the price is updated whenever a new price arrives to the market. Two consecutive ticks can arrive within microseconds of each other. On the other hand, consecutive ticks can also be separated by long periods of time, which implies that no data is recorded for the duration. Although equidistant data is not necessary for the HMM the analysis is simplified in this case. Furthermore, as explained above, the duration between consecutive ticks does not imply that there is missing data, rather it implies that the price has not changed since the last observation.

In order to obtain equidistant time points, the data was sorted and arranged with a sampling rate of 1 second, by setting the price at each second equal to the last observation. Naturally, this method is insensitive to price variations in shorter time scales then the sampling rate, but it still offers a good approximation of the price process while adding the simplicity to the modelling. One important note is that the data does not contain any transaction information.

Somewhat visible in the data, is that the price processes have periods, of varying length, where they are constant. This implies that the absolute return is zero for a substantial fraction of the observations. In fact, the zeros constitute nearly 40% of the observed absolute returns. This is a clear motivation for the use of zero-inflated models in the analysis.

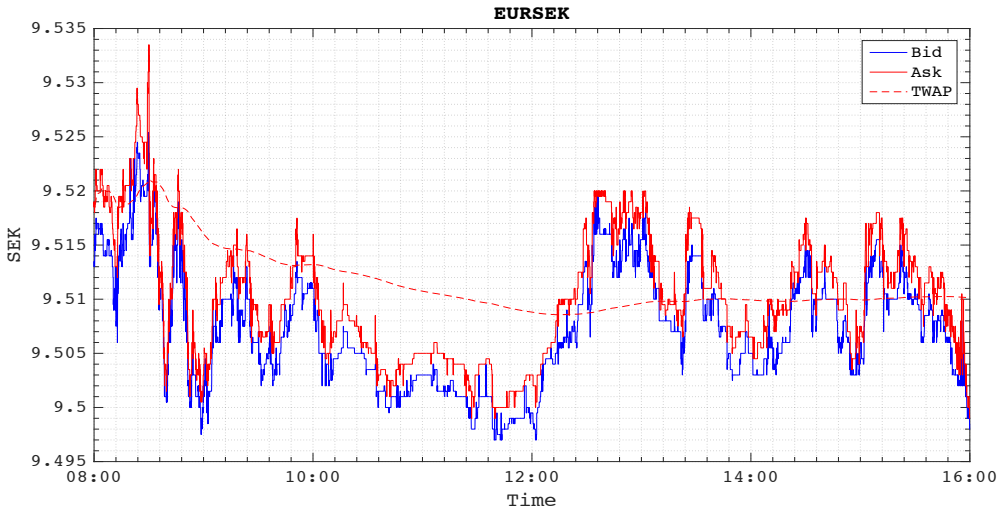

Figure 3.1: *Plot of the EURSEK during one European trading day (2017-01-12), where time is expressed in GMT. The plot shows the best bid and ask during the day, together with the TWAP, equation (4.1), of the ask prices.*

3.2 Intraday variations

The behaviour of the price process varies over the course of the trading day, according to the market activity, which in turn depends on many different factors. The rate of change in price is directly proportional to the market activity, or the number of participants in the market place. A higher rate implies larger activity in the market, which is a consequence of having more participants. The figure below shows the average daily turnover, measured over a month, which is the percentage of the total daily volume traded per 30 min interval over the day.

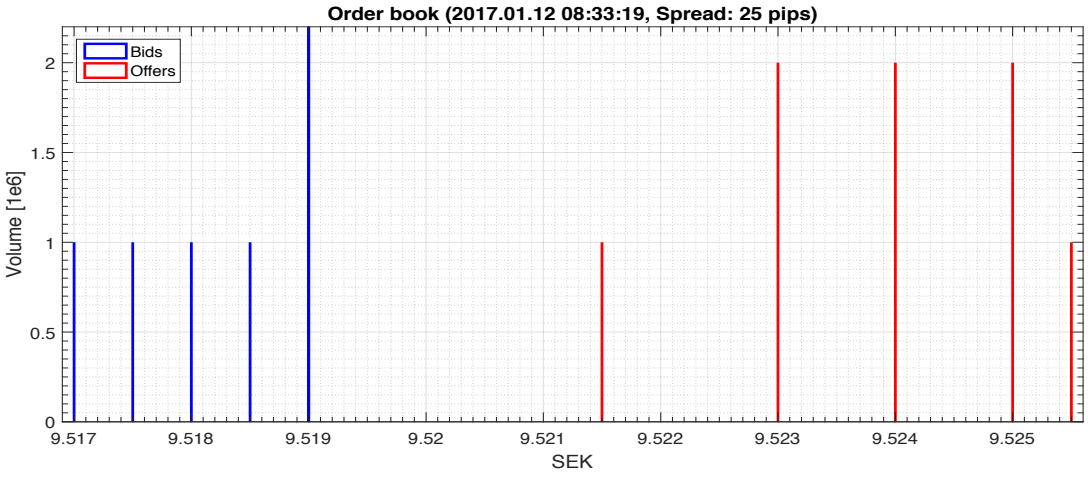

Figure 3.2: *Snapshot of the order book.*

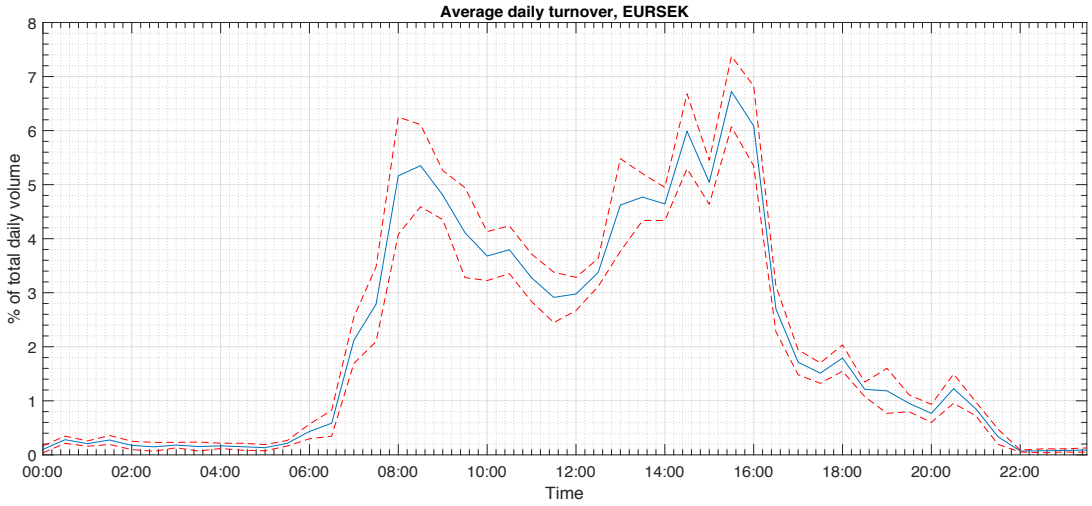

Figure 3.3: *Average turnover for EURSEK, as a function of time. The blue line is the mean turnover, calculated using daily values of the turnover measured between 2017-01-04 and 2017-02-11 (29 trading days), at each time and the red dotted lines show two standard deviations. The area under the curve sums to 1.*

As indicated in the figure, there are 2 large peaks in the trading activity, which correspond to overlapping operation hours for financial centers where FX is traded (see figure 1.1). The first peak corresponds to the opening of the London market, which is one of the world's largest [46]. After the initial peak, there is a decrease in activity over the day, with the minimum value occurring around lunch time. The activity then increases

during the afternoon, in anticipation of the New York market and reaches another peak approaching the closing times of the European markets, after which there is a large drop-off in activity.

Chapter 4

Trading

4.1 Trading factors

Several factors affect the trading of FX, including all steps from what kind of order to place, to assessing the risks in the trading to evaluating the results. The most important factors are explained and described in the following section. The second part of the chapter defines the trading strategy and explains the rationale behind some of the assumptions made in the modelling.

4.1.1 Order types

There are essentially two types of orders that can be made on the FX-markets: market orders and limit orders. A market order is simply an order to trade a specified quantity at the best price possible in the market. The purpose of the market order is to quickly perform a trade, without no price limit in mind. Market orders consume liquidity, with a buy market order trading at the best ask price and a sell market order trading at the best bid price. The price paid for the immediate execution, generally speaking if the order can be filled, is the spread between the best bid and best ask.

The second type of order is the limit order. A limit order is an instruction to trade a specified quantity at a specified price. If the price limit specified in the order can not be met, the order simply sits in the book until either the market price reaches its limit or it's cancelled by its placer. Limit orders provide liquidity by creating limit buy and limit sell orders, thus creating a market. The reward for providing liquidity is that the placer of the order is allowed to specify the price. This yields the spread, between limit orders on buy and the sell side of the book.

Limit orders have 2 parameters: the limit price and the quantity. A limit order buy(sell) can be placed anywhere between the worst and best bid(ask) in the book, with aggressive

orders closer to the best price (top of the book) having a larger probability of execution. Orders for large quantities may sit on the book for a longer time as they are more difficult to fill and are partially filled until the volume is depleted.

Limit order are versatile and can be greatly tailored to the traders need. Instructions on the duration, fill behaviour, exchange routing and more can be set for the limit order, together with cancellation at any desired time.

In this thesis, focus will be on market orders and simple limit orders, meaning that a limit order is available on the book until it's either completely filled or cancelled.

4.1.2 Benchmarks

To monitor and assess the performance of trades and executions of orders, price benchmarks are used in FX trading. Many types of benchmarks exists, depending on the preferences of the trader, analyst or client. For example, opening and closing price are often used as benchmarks for "profit and loss calculation" [19, p. 48]. Intraday benchmarks use average prices over the day, which more accurately reflect market conditions. The most common intraday benchmarks are the time-weighted average price (TWAP) and the volume weighted average price (VWAP).

As the name suggests, the TWAP is simply the moving average of the best prices at the market over day,

$$TWAP(T) = \frac{1}{T}\sum_{t=1}^{T} P_t. \tag{4.1}$$

The TWAP is a somewhat naive or simple benchmark, as it doesn't provide any insight on relevant market conditions, such as volatility and liquidity. But it's an approximation of the price level of the market over the day, giving an estimate of the expected price if trading is performed over the day. The TWAP can be calculated in real time as follows,

$$\begin{aligned} TWAP(T) &= \frac{1}{T}\sum_{i=1}^{T} P_i \\ &= \frac{1}{T}\left(P_T + \sum_{i=1}^{T-1} P_i\right) \\ &= \frac{1}{T}P_T + \frac{T-1}{T}TWAP(T-1), \end{aligned}$$

for $T \geq 1$.

Note that the buy(sell) TWAP is calculated using the best ask(bid), meaning that it's calculated based on trades that can be executed immediately.

The VWAP weights each trade with the price and volume, offering a more practical measure of performance, explaining its popularity among almost all other asset classes[19, p. 49],

$$VWAP(T) = \frac{\sum_{t=1}^{T} P_t Q_t}{\sum_{t=1}^{T} Q_t}. \tag{4.2}$$

The VWAP can also be calculated in real-time as follows,

$$VWAP(T) = VWAP(T-1) + \frac{P_T Q_T}{\sum_{t=1}^{T-1} Q_t + Q_T}.$$

The role of benchmarks in algorithmic trading is often to evaluate total transactions and execution strategies, by allowing the trader and client to compare the performance of the trade and the benchmark.

The main reason for using TWAP instead of VWAP is lack of volume data, which is the case for FX-markets. Traded volumes are not disclosed, hence it's not possible to calculate the VWAP. Hence, VWAP can only be used to assess the effective price obtained through a strategy after the trading is completed, as opposed to assessing the current state of the market.

4.1.3 Trading

Trading is performed through traders placing market or limit orders on venues. This can be done on the traders own account or on behalf of a client. In the example of a bank offering currency exchange services, clients commission the bank to perform the trading on the client's behalf. There is an economical incentive for the client to commission the bank to perform the trading on the client's account. The bank has access to inter-bank markets, which generally offers better exchange rates compared to public markets. Furthermore, the bank is allowed to place limit orders (i.e. trade as a market maker), while the client generally only can place market orders (i.e. trade as a market taker). That is, if a client desires to buy a currency, the client can do so by lifting offers(asks) of the market. The bank, on the other hand, can buy the desired quantity by placing limit buy orders on the bid side of the order book.

The bank, or trader, can trade large volumes in mainly 2 ways: the TWAP and VWAP(or POV). In the TWAP algorithm, the trader simply places equally sized and equidistant market orders until the target volume is achieved. In the VWAP algorithm the trader places limit orders, with size and time depending on market conditions, until the target volume is achieved. The advantage of the VWAP is that the spread is not crossed, which

Trading Costs	Explicit	Fixed	Implicit	Variable
Commission	x	x		
Fees	x	x		
Market Impact			x	x
Timing Risk			x	x
Price Trend			x	x
Spread			x	x
Opportunity Cost			x	x

Table 4.1: *Table showing some of the important trading costs and risks, together with their nature.*

reduces the cost. On the other hand, due to uncertainty in the execution time, the VWAP algorithm has a larger timing risk as it has a large exposure to market volatility. As such, both algorithms have advantages and drawbacks and a decision between the two is made using the client's preferences.

4.1.4 Risks

Various factors affect the cost of trading. They are summarized in table 4.1, also showing their nature.

Any trading incurs commissions and fees for the trader, which may or may not be the broker as well. The total trading income, for the trader, per unit of currency traded can be summarized as follows

$$TradeInc = S\left[Inc - \left(Broker + \frac{1}{s}Settl\right)\right], \tag{4.3}$$

where S is the total volume, Inc is the income per unit currency, $Broker$ is the amount paid to the broker per unit currency, $Settl$ is the settlement cost per trade and s is the size of each trade. All of these parameters are constant. The total trading income is higher for fewer trades, on the other hand, the market impact is increasing in larger trade sizes. Hence, to maximize trading income, a trade off must be made between the market impact and settlement cost.

The spread cost is the difference between the best bid and ask at any time. It is compensated to those who provide liquidity and paid by those who consume liquidity.

Opportunity cost is associated with the cost incurred when an order is not executed. Regardless of the reason for why the trading was not completed, the remaining volume failed to trade and is thus subject to all the risks mentioned in table 4.1.

The timing risk is the risk associated with the duration of the market exposure. It is mainly affected by the price risk and the liquidity risk, but can include other factors. Specifically, the price risk measure the volatility exposure for the remainder of the trading time. It can be estimated as follows [19, p. 300]

$$TR = \rho_0 \sqrt{\sum_{j=1}^{n} r_j^2 \cdot \frac{t\sigma^2}{n}} \tag{4.4}$$

where r_j is the residual position in bucket j, $t\sigma_j^2/n$ is the volatility in each bucket and ρ is a scaling constant.

The liquidity risk concerns the variability in market activity and liquidity over the trading periods, which also affects the market impact. It can be estimated using historical data or liquidity models, see for example Kissell et al (2004).

Market Impact

Market impact refers to the effect of the trader's own actions and orders on the market price process. Much effort has been put into modelling market impact, most notably the framework developed by Almgren and Chriss (2001). In this framework, the total market impact of trading consist of two parts: a temporary impact function and a permanent one. As the name implies, the temporary impact function represent the immediate effect of an order and its diminishing effect over time. The permanent impact function refers to the lasting effect of the order on the price process. Various functional form and estimation procedure are also developed in Almgren and Chriss (2001), but no consensus. This is an indication of the difficulty of modelling market impact.

4.2 Strategy

4.2.1 Objectives

For any form of successful trading, a sensible and profitable trading strategy is required. A strategy, in turn, is devised based on set objectives. The strategy developed in the following is derived to achieve the following objective:

> *Trade a large volume, split up over the day, so as to achieve the best possible VWAP compared to benchmarks, while taking into consideration trading risks, market conditions and client preferences.*

Note that the buy and sell cases are symmetrical and the buy case will be under study in the rest of this thesis. In this case, "Trade" in the sentence above becomes "Buy" and "Best price" become "Lowest price". Benchmarks here refers to the market TWAP during the trading time. The risks are the ones described in the previous section.

To achieve the objective stated above, some questions must be answered by the strategy and they are given in table 4.2.

Macro decisions	**Micro decisions**
-How to slice?	-Market or Limit?
-When to trade?	-Aggressiveness?
-Size of slice?	-Fill instructions?
-Trade horizon?	
-Risk aversion?	

Table 4.2: *Trading questions for a strategy.*

Most of the questions are fairly intuitive for an order splitting strategy. The trade horizon is related to the risk aversion through the client preferences or investor criteria. For example, there could be constraints on the time of completion for the trade.

The trader can choose between placing market or limit orders, which both have their respective advantages and drawbacks. Aggressiveness refers to the price, compared to the best bid, at which a limit order is placed. Limit orders placed at the best price are the most aggressive. Fill instructions are specifications for the limit orders.

The risk aversion indicates how sensitive the client is to risk. A risk averse client dislikes risk and rather pays a premium to reduce the risk. In this case, the premium is the increased VWAP due to the market orders used in the trading, also allowing the trading to be completed faster as execution of orders are immediate. That is, the risk averse client pays the spread, but in turn will receive a VWAP that is close to the market TWAP. On the other hand, a risk inclined client is less sensitive against risk and allows for a longer trading horizon with the hope of obtaining a better VWAP using limit orders.

A viable trading strategy answers all the questions in table 4.2, as well as incorporating real-time market conditions together with client preferences and criteria. Such a strategy will be described in the following section.

4.2.2 A Hybrid Limit-Market Strategy Framework

The proposed strategy framework is a hybrid between a VWAP algorithm and a TWAP algorithm. The strategy divides the total volume between the two algorithms and then places limit and market orders to achieve the trading goal. That is, if the total volume is denoted by S the volume is then distributed as follows

$$\begin{aligned} S =& S(1-\alpha) + S\alpha \\ \triangleq& S^d + S^T, \end{aligned} \tag{4.5}$$

where α is the risk aversion of the client. The strategy then contains the VWAP($\alpha = 0$) and TWAP($\alpha = 1$) as special cases. The TWAP part of the strategy trades according to the TWAP algorithm, with volume and trading horizon set by the VWAP part of the strategy. The VWAP part of the strategy, on the other hand, uses some different parameters to decide how to place limit orders, and will be explained below.

The engine of the VWAP part (referred to as the d-strategy in the following) uses a volume distribution function to inform trading decision. Assume, for the moment, that the rate at which limit orders are filled is known as a function of order volume and market activity. This rate can then be used to estimate the trading horizon, for a given S^d, for the d-strategy. Using this horizon, the volume S^d is then distributed over intervals, or buckets, over the trading day using historic market data on the daily average turnover as a function time. This distribution then indicates how much volume to trade, using limit orders only, in each bucket. Larger turnover for a bucket implies a larger volume to trade. Using local predictions from a price model, limit orders are then placed in each bucket until either the volume for the bucket is depleted or the end of the bucket is reached. The volume distribution is then updated and redistributed using the information from the last bucket and the time left on the trading horizon. The d-strategy then continues in this way until trading is completed. Using this trading scheme, benefits of market and limit orders are combined in trying to obtain a better overall VWAP, while achieving the objectives.

The volume distributor should depend on several factors. Let $d()$ denote such a volume distribution, giving how much volume to trade in the current bucket. A general form for it is given in the equation below

$$d = d(t, S^d, S^T, \alpha, Turn, MI, TR, SVWAP - TWAP), \tag{4.6}$$

where the parameters are defined as follows:

- S^d is the volume left for the d-strategy. The d-function is dynamic in that it updates the volume distribution depending on how much is left of the total volume. $d()$ is increasing in S^d.
- S^T is the volume left for the TWAP part of the strategy. The d-function redistributes volumes depending on the progress for both parts of the strategy. $d()$ can be increasing or decreasing in S^T.
- α is the risk aversion of the client. The d-function depends on α through the market impact and the timing risk, with a higher α trying to reduce the timing risk. $d()$ is increasing in α.
- $Turn$ refers to the turnover of the market. The volume is distributed according to historical volume profiles, but can be updated using real-time measurements of market activity. $d()$ is increasing in the turnover.
- MI is the market impact. Market impact is generally undesirable as it can nega-

tively affect prices and reveal the intentions of the trader to other market participants. $d()$ is generally bounded by the market impact.

- TR is the timing risk and refers to the risk associated with the market exposure. A larger order requires a longer trading horizon, which in turn implies a larger exposure to market volatility. $d()$ is increasing in the timing risk. TR is a function of α.
- $SVWAP - TWAP$ refers to the current VWAP of the total strategy (SWVAP) and the current TWAP of the market. This parameter allows for adjustments of the volume distribution depending on how the strategy is performing. For example, if the performing badly, more volume can be distributed to the VWAP part of the strategy, and vice versa. $d()$ is decreasing in this parameter.

Note that all parameters are functions of time. The $d-$function states how much volume to passively trade in the current bucket. It is updated at the end of each bucket to incorporate past events and future conditions.

Such a $d-$function then answers most of the questions in table 4.2, only leaving the questions of when to trade and how aggressive the limit orders should be. The question of when to trade can be answered using trend predictions from a price model. Trends are predicted using a price model and are then used to make short-term decisions. For example, if the predicted trend is negative (i.e. the price will decrease), the decision could be to not place a limit order for the duration of the prediction horizon. If the predicted trend is zero, the decision is to place a given quantity for the limit order. Finally, if the predicted trend is positive, the decision is to place a limit order for larger fraction of the given quantity. For example, the trend predictions $-1, 0, 1$ could correspond to limit orders of quantity $0, 1, 2$ volume units. This explains the usefulness of the precision measure defined in equation 2.3. That is, placing orders of quantity 1 or 2 is still good, as long as the true trend is not negative and this is the penalty defined in equation 2.3.

The aggressiveness of the limit order affects its execution probability, with larger and less aggressive orders requiring longer time to find matches on the market. Based on the execution probability as a function of order size and aggressiveness, together with the market activity, it is then possible to find the most profitable setting.

A strategy framework as the one described above offers a balance between the benefits of both market and limit order. Its dynamic updating allows it to react to current market conditions and adjust its behaviour accordingly. It also allows for the trading to be tailored to the clients preferences. Although exact results for the performance can not be derived without assumptions for the price and volume distributions of the market, some postulations can be made for the strategy. Its performance should, ceteris paribus, not be worse than that of a TWAP algorithm. Its performance compared to a VWAP strategy depends on the market conditions. The strategy will complete the trading in a shorter time period compared to a pure VWAP strategy. This implies that the pure VWAP has a larger timing risk, which can produce better or worse performance

depending on market conditions. However, the relevant benchmark for the algorithm should be a trading strategy executed in the same settings, including the trading horizon.

The strength of the modular formulation of the strategy is that each parameter can be modified separately, without altering the rest, allowing for many different strategies to be contained within the proposed framework. For example, the timing risk can be calculated in many different ways, but this does not affect the $d-$function. In a similar manner, the strategy framework allows for any functional form of the $d-$function. Using the notes on how each parameter should affect the $d-$function, a suitable form can be derived.

4.2.3 Simplifications & Simulations

To offer some indication on the performance capacity of the strategy framework presented, a simple version, adhering to the previous section, is presented in the following. Specifically, the following assumptions are made:

1. S^d, S^T, α and $Turn$ are known
2. MI is set to be constant, independent of market activity. It is a fraction such that the maximum allowable trade volume per bucket is a fraction of the total market turnover
3. TR is set to be constant, independent of market volatility and the remaining position, and additive to the $d-$function

In this case, the $d-$function has the following form

$$d_t = MI * v_t + TR, \tag{4.7}$$

where v_t is the turnover, in millions, at time t. The $d-$function is therefore linear in the turnover, market impact and timing risk.

Execution probabilities can be obtained from market data on the lifetimes of limit orders. Such data was unavailable at the time of writing, leaving the executions probabilities to be estimated from the data on average number of trades, as a function of time. It's assumed that the execution probability is an increasing function of the number of trades. This assumption can then be used as a proxy for the execution probabilities. First, note that trades can either be limit or market orders. Assuming that the distribution between the two is constant over the day, the uninformative guess is that the trades are equally distributed between the two. Second, note that limit orders can be filled, partially filled or cancelled. Again, assume that the distribution between them is constant as a function of time. Similarly, the best uninformative guess for their relative distribution is uniform. Thus, an estimate for the amount of filled limit orders, as a function of time, is obtained. The time dependence follows from the average number of trades, which are measured each 30 min period over 60 trading days. Hence, the obtained estimate is the average

number of filled limit orders per 30 minute, or per bucket. Lastly, assuming that limit orders are filled at a constant rate and independent of each other, the number of limit orders follows a Poisson distribution. Let λ be the rate per bucket and t the length of a bucket. The Poisson then has mean λt and the time between orders being filled then follows a $Exp(\lambda)$ distribution. The total time required for N limit orders to be filled is then given by the gamma distribution, or

$$\sum_{i=1}^{N} X_i \sim \text{Gamma}(N, 1/\lambda),$$

with mean $N\lambda^{-1}$. The rates λ are different for each bucket. An estimate of the trading horizon for the strategy can now be obtained as follows

$$T\lambda s = S^d \Rightarrow \ T = S^d/\bar{\lambda}s$$

where $\bar{\lambda}$ is the average execution rate and s is the average size for limit buy orders. The left hand side in the first equation is the expected value of a Poisson with rate parameter λ, multiplied with the average size of each trade. This horizon can then be used for the TWAP part of the strategy. Together with trend predictions from the price model, the trading strategy can now readily be implemented.

Chapter 5

Modelling

5.1 Price model

The proposed price model, based on the ZIP-HMM, under study in this thesis can be written down as follows:

$$P_{t+1} = P_t + Ca_t, \tag{5.1}$$

where P_t is the price at time t, C is a scaling constant equal to the magnitude of one pip and $\{a_t\}$ follows the distribution induced by the HMM. From this equation, it is easy to see that it is the absolute returns of the price that are studied, or

$$a_t = \frac{P_{t+1} - P_t}{C}.$$

a_t has the same unit as the price, as C is dimensionless. Time is measured in seconds such that there is one second between consecutive observations at $t + 1$ and t.

5.2 Model training

5.2.1 Initial Parameter Estimates

The log-likelihood surface of the HMM is a function of the data, the dimension of the HMM and the form of the observation distribution, which in practice means that these surfaces are generally highly complex. Therefore, there is no straightforward way to infer where, e.g. at what parameter values, maxima of the log-likelihood occur, nor is it simple to determine whether the extreme values correspond to local or global maxima. The EM-algorithm is guaranteed to converge to local maxima. As such, the most common way to explore the log-likelihood surface is simply to run the algorithm using different initial estimates for the parameters.

The mixture components, the transition matrix and the initial distribution are all subject to stochastic constraints in order for them to form proper distributions. Hence, initial estimates can be obtained by simply generating random vectors and matrices, as no other information about their form is available, and properly normalizing them [39, Section. 5C]. The lambda parameters of the Poissons are only subject to a positivity constraint but the role of the parameter in the distribution provides additional information about its effect. The expected value of a $Po(\lambda)$ distributed random variable is λ, hence initial values for the lambda parameters are obtained by setting them to be the means of clusters in the data.

The number of clusters is determined by the dimension of the HMM, that is the number of states for the chain and the number of Poissons in the observation distributions. The clusters themselves can be found using only the EM-algorithm but as it is sensitive to the initial parameter values, both in terms of the rate and stability of the convergence, the K-means algorithm is commonly used to find clusters in the data and calculate the sample mean of each cluster. The K-means algorithm is implemented in MATLAB through the `kmeans` function. Initial estimates for the HMM are therefore produced by generating random normalized vectors and matrices, together with the cluster means found by iterating the K-means algorithm for a few steps. To further promote exploration of the log-likelihood surface, variation can be introduced in the K-means estimates by only using a randomly selected sub-sample of the data when running the clustering algorithm, producing different clusters for each run.

The log-likelihood surface is difficult to illustrate as it is a function of many parameters. Some indication of the location of maxima can nonetheless be obtained by studying the convergence of the parameters. This will be analyzed by plotting parameter trajectories as functions of the number of iterations made in the EM-algorithm.

5.2.2 Stopping Criteria

Stopping criterion are necessary for the EM-algorithm in order to terminate when the algorithm appears to have found a maximum of the log-likelihood surface. Also, stopping criterion can prevent the EM-algorithm from converging to singularities of the log-likelihood function and spurious local maximizers, which could correspond to mixture components collapsing onto one data point [29, p. 99]. A common way to monitor convergence is to record the change in log-likelihood between subsequent values. If the difference is below some threshold, or if the number of iterations have reached the maximum allowed, the iteration stops. The same value for the threshold and the maximum allowed iterations were used in all runs of the EM-algorithm, set to 10^{-6} and 300 iterations.

5.2.3 Training data

The training data used as input in the EM-algorithm are the best bids on the market, observed between 08:00 and 16:00 GMT. These time periods correspond to the highest market activity on the European markets. The price process for the best asks on the market is essentially identical to the bids, with some variations due to fluctuations in the spread over the day.

The currency price data is discrete in the sense that the smallest unit of variation, pips, have a fixed size compared to the price. As the zero-inflated Poisson model is used to model pips, i.e. the model input is the absolute return in pips, the data needs to be transformed. This is done by simply calculating the change in price between subsequent observations and scaling the result with the inverse of the magnitude of one pip. This data will be referred to as count data.

The Poisson distribution only has support on the non-negative integers, while the count data contains negative integers, corresponding to a decrease in price. It is possible to incorporate a translation of a Poisson random variable by adding a constant c and then treat the constant as a parameter of the distribution, i.e.

$$X \sim Po(\lambda), \quad Y \triangleq X - c, \quad p(Y = k) = p(X - c = k) = \frac{e^{-\lambda}\lambda^{k+c}}{(k+c)!}.$$

Maximizing this distribution, which is necessary in the M-step of the EM-algorithm, with respect to the parameter c becomes problematic due to its appearance in a factorial. It might be possible to circumvent this problem by running the generalized EM algorithm [34], which does not maximize the complete data log-likelihood at each iteration, but instead tries to change the parameters such that the log-likelihood increases [7, p. 454]. This approach, however, was not further investigated in this thesis and the count data was simply translated, such that all value where non-negative, before being used in the algorithms. The effects and consequences of this method will be further analyzed in the discussion.

5.2.4 Simulation study

To study the convergence properties of the algorithm and the behaviour of the ZIP model, the HMM will first be trained using simulated data with known parameter values. That is, the data generating process is a ZIP(2,2) model. The sensitivity of the EM-algorithm to the initial values will also be assessed using a plot of the convergence trajectories for the emission distribution parameters.

It would be of interest to have a measure of similarity or distance between HMMs, in order to quantify how well the estimated models replicate the true model as well as offering a bound for the expected performance. Also, as noted in [39], even though two HMMs appear to be different, with different parameter values, they can still be equivalent

in a statistical sense. For example, the conditional expectations in the ZIP(2,2) model involves more than 10 parameters, which implies that many models can produce the same distributional properties.

In [20] the authors proposed a "probabilistic distance measure for measuring the dissimilarity between pairs of hidden Markov models with arbitrary observation densities". The measure is based on a limit theorem from [37]. Specifically, let Θ denote a probabilistic model, including a transition matrix A and observation probabilities B, defining a measure denoted by $\mu(\cdot|\Theta)$. Furthermore, let $O_{1:T}$ denote an observation sequence of an ergodic stochastic process, from time 1 to T, generated from the measure $\mu(\cdot|\Theta_0)$, and define the function

$$H_T(O,\Theta) = \frac{1}{T}\log\mu(O_{1:T}|\Theta),$$

for each T and every observation sequence $O_{1:T}$. $H_T(O,\Theta)$ is a random variable of the probability space of models Θ. The limit theorem in [37] proves the following limit

$$\begin{aligned}\lim_{T\to\infty} H_T(O,\Theta) &= \lim_{T\to\infty}\frac{1}{T}\log\mu(O_{1:T}|\Theta)\\ &= H(\Theta_0,\Theta),\end{aligned} \tag{5.2}$$

where the limit exists almost everywhere $\mu(\cdot|\Theta)$. The theorem also proves the following inequality

$$H(\Theta_0,\Theta_0) \geq H(\Theta_0,\Theta), \tag{5.3}$$

with equality if and only if Θ is in the set of probability models such that $\mu(\cdot|\Theta) = \mu(\cdot|\Theta_0)$, i.e. Θ is in the set of probability models that are indistinguishable by the probability measure $\mu(\cdot|\cdot)$. Using these results, the following distance measure can be defined

$$\begin{aligned}D(\Theta_1,\Theta_2) =& H(\Theta_0,\Theta_0) - H(\Theta_0,\Theta)\\ =& \lim_{T\to\infty}\frac{1}{T}\big[\log P(O^{(2)}_{1:T}|\Theta_2) - \log P(O^{(2)}_{1:T}|\Theta_1)\big],\end{aligned} \tag{5.4}$$

where $O^{(2)}_{1:T}$ is a sequence generated by the model Θ_2, and $P(\cdot|\Theta_i)$ are measure induced by the probabilistic models Θ_1 and Θ_2, respectively. The distance measure has information theoretic interpretations and 5.4 can be proven to be the Kullback-Leibler number between the two measures $P(\cdot|\Theta_2)$ and $P(\cdot|\Theta_1)$. More details on the derivation of the distance measure and its behaviour for different models can be found in [20].

In order to quantitatively assess how well the models estimated through the implemented EM-algorithm appear to resemble the generating model, the distance between the models will be calculated according to equation 5.4. The distance measure on simulated data also indicates how much data the model needs to converge and, once it has converged, it suggest a lower bound of the error or distance for the estimated model from the true model. A lower bound since the distance obtained in the ideal setting where the generating and estimated model are of the same form and dimension.

5.2.5 Implementation

All of the calculations and algorithms were implemented using MATLAB [27]. Some parts of the EM and BW algorithms do not depend on the form of the observation distribution and were implemented using functions from the MATLAB toolbox for HMMs by Murphy [32]. The remaining calculations were vectorized to the largest extent possible in order to reduce computation time by exploiting MATLAB's efficient operations for vector and matrix operations.

5.3 Model fit

This section describes the methods used to analyze the fit of the HMM on the data.

5.3.1 Dimension of the HMM

Information criteria are used to assess and compare the fit of models with different dimensions, trained on the same data set. The two most commonly used are the Akaike information criterion, defined as

$$AIC = -2\log L + 2p, \tag{5.5}$$

and the Bayesian information criterion, defined as

$$BIC = -2\log L + p\log(T), \tag{5.6}$$

where $\log L$ is the log-likelihood of the model on the data, T is the number of observations and p is the number of parameters in the model. From these equations it is easy to note that the BIC penalizes more complex models heavier than the AIC for $T > e^2 \approx 8$, which holds in almost all applications. The BIC therefore favours simpler models compared to the AIC [51]. Both of these ICs have the same form, with the first term measuring the fit of the model. It's decreasing with the number of parameters. The second term is the penalty term and increases with the number of parameters.

The ICs are calculated for all estimated models and the model with the lowest IC (AIC or BIC) is the preferred one. As such, it is the difference in the IC-values that is of importance when comparing models. These differences can be interpreted in terms of probability of information loss, but if suffices to note here that $\Delta IC_i \triangleq IC_i - IC_{min} > 10$, where IC_i and IC_{min} are the ICs for the i :th model and the best model, is enough to dismiss model i [3].

The ICs can also be used to calculate posterior probabilities of models. Let $\{m_i\}_{i=1}^{M}$ denote a set of models and π denote the prior distribution over the models. The Bayesian

posterior probability[3] for model m_i is then given as follows

$$P(m_i|\text{Data}) = \frac{\pi_i \exp -\frac{\Delta\text{BIC}_i}{2}}{\sum_{j=1}^{M} \pi_j \exp -\frac{\Delta\text{BIC}_j}{2}} \tag{5.7}$$

Models with lower BIC values have larger weights in this distribution, with largest weight assigned to the model with the smallest BIC value. With no prior information as to the relevance of each model, the prior can be set to be the uninformative uniform distribution.

Although work still remains to be done in the analysis of order estimation of HMMs, the BIC has been proven to be a strongly consistent Markov order estimator, which is a most desirable property for a good estimator [9]. Hence, the BIC will be the preferred IC in this thesis.

5.3.2 Learning Curves

The HMM is initially estimated using 8 hours of data (08:00-16:00), corresponding to the period of the day with the most market activity. Different combinations of the number of states and mixture components, according to table 5.1, are used and the resulting log-likelihoods are stored and used to calculate the AIC and BIC, from which the dimension of the preferred model can be found.

	1	**2**	**3**	**4**	**5**
1	1	3	6	8	10
2	12	16	20	22	24
3	21	27	33	39	45
4	32	40	48	56	64
5	45	55	65	75	85

Table 5.1: *The number of parameters to estimate in the EM-algorithm as a function of the number of states (leftmost column) and the number of Poissons mixture components in the observation distributions, excluding 1 Dirac component.*

For real-time applications, it is important that the model does not require a prohibitive amount of time and computation to produce reliable estimates. Furthermore, given the rapidly changing conditions in the market, yesterday's data is generally a bad predictor of the market today in HFT. Consequently, it is of interest to study the HMM's performance, measured as prediction accuracy, as a function of the number of training data points. Specifically, the HMM is trained using increasing lengths of the training data sequence, corresponding to increasing time intervals, during 3 different times of the day. The dimension of the HMM is obtained from the previous analysis on the full

data set. The best parameter values, i.e. the parameter estimates from 10 runs of the EM-algorithm on the data, are stored and plotted as a function of the length of the training sequence, forming the LCs. With the help of the LCs, the trade-off between prediction accuracy and computation cost, which is highly dependent on the size of the training data, can be assessed.

To reduce the possibility that the HMM trained on a full day of data is insensitive to intraday variations, the optimal model, with respect to the size of the training data, from the LC analysis will be evaluated at different times of the trading day under different market conditions. The ICs will then be used to investigate if any dimension of the HMM is preferred, compared to the dimension found for the full data, as described in the previous section.

5.4 Model Performance

This section describes the methods used to evaluate the prediction performance of the HMM on the data.

5.4.1 Price Prediction

The intended use of the model is for predictive modelling, hence predictions must be obtained from the model. Predictions of the price are obtained from the predictive distribution, as described in the theory section. The prediction accuracy of the HMM is assessed by calculating the mean prediction error (MPE) and the standard deviation of the prediction error (SDPE), defined as follows

$$MPE_t \triangleq \frac{1}{M}\sum_{j=1}^{M}(y_t - x_{tj}), \tag{5.8}$$

$$SDPE_t \triangleq \sqrt{\frac{1}{M}\sum_{j=1}^{M}(y_t - x_{tj})^2}, \tag{5.9}$$

where y_t is the observation at time t and x_{tj} is the j:th prediction at time t.

After the MC has converged to the stationary distribution, say when $s > n$ for some $n \geq 1$, the prediction intervals will remain constant for all $s > n$, generating constant predictions independent of time which is unrealistic. However, the predictions of the HMM are probably still relevant up to some time bound, after which they become unreliable. Therefore, it's of interest to determine for how long the predictive distribution appears to be valid. That is, how long the market does appear to follow the model before it needs to be re-calibrated using new data, is of relevance. This will by studied by calculating the MPE and SDPE for the HMM using different prediction horizons during

different times of the day. The prediction accuracy of the HMM will also be compared to the Geometric Brownian Motion of financial time series.

5.4.2 Trend Prediction

While prediction error, as calculated above, is a common way to gauge the performance of a predictive model, the utility gained from predicting prices is not always clear. For example, Buy-and-Hold strategies rely on predictions of the direction of price movements, as compared to the exact price, for trading decisions. Intuitively, predicting the trend in price process should be easier than predicting the exact price, as the latter is an outcome from a much larger probability space compared to the former. That is, the future price can increase, decrease or remain constant and nothing else, while the exact price can take numerous values. Obviously, a model able to accurately predict future prices will also predict trends well. But a model with poor accuracy in price predictions might prove to be useful if it can accurately estimate trends and offer more insight than simply randomly guessing the future trend. Hence, the HMM's ability to predict the direction of the price process is of importance and will be studied.

Classification

The study of the trend prediction can conveniently be cast into a classification framework. Specifically, let P_t denote the price at time t and $C_{t,t+s}$ ($s \geq 1$) the true class at time $t+s$ relative to time t. The classifier can now be defined as follows

$$C_{t,t+s} = \begin{cases} -1, & \text{if } P_{t+s} < P_t - \varepsilon \\ 0, & \text{if } P_t - \varepsilon \leq P_{t+s} \leq P_t + \varepsilon \\ 1, & \text{if } P_{t+s} > P_t + \varepsilon \end{cases}, \tag{5.10}$$

where ε is a variable allowing for some slack. The trend predictor, or classifier, on the other hand is not as straightforward to define. One way to classify the observations is through the Bayes' classifier. It simply assigns, for each observation, the class that is most likely [18, p. 38]. It is well known, in the classification setting, that the error rate, the average number of miss-classifications, is minimized by the Bayes' classifier. The problem is that the distribution of the each class conditional on the data is required, which is unknown unless the true distribution of the data is known.

Another possible way to define the classes is through the cumulative distribution function of the generated predictions. That is, the exact form of the predictive distribution $P(O_{t+s}|O_{1:t})$, as given in equation 2.40 in the theory section is known. In the section on mixture distributions it was demonstrated how the CDF can be derived in equation 2.9. Using this, together with the realization that $P(O_{t+s}|O_{1:t})$ is a mixture of mixture distributions, allows for the CDF to be calculated. By studying the location of the last observations compared to the CDF, predictions are then made about the apparent

trend in the data. Specifically, let X be a random variable with CDF $F(x)$ and let $Q(p) = \inf\{x \in \mathbb{R} : p \leq F(x)\}$ be the quantile function. The following quantiles of the CDF are then calculated

$$LP = Q(p - \delta),\ CP = Q(p),\ UP = Q(p + \delta). \tag{5.11}$$

Using these quantiles, the following distances can then be calculated

$$d_1 = |O_t - LP|,\ d_2 = |O_t - CP|,\ d_3 = |O_t - UP|. \tag{5.12}$$

Now, let $d = [d_1, d_2, d_3]$ and let $\hat{C}_{t,t+s}$ denote the estimated class at time $t + s$, relative to time t. The classifier can now be defined as follows

$$\hat{C}_{t,t+s} = \begin{cases} -1, & \text{if } \min_i d(i) = 1 \\ 0, & \text{if } \min_i d(i) = 2 \\ 1, & \text{if } \min_i d(i) = 3 \end{cases}, \tag{5.13}$$

In words, the classifier assigns a class based on the distance from the last observation to the 3 points in equation 5.11, which give information on the form of the CDF.

Ensemble Classifier

As mentioned in earlier sections, the parameter estimation is sensitive to the initial estimates in the BW-algorithm. Several runs of the BW-algorithm are therefore used to produce, hopefully, better parameter estimates. It is possible to only use the estimates from the best model, i.e. the one with highest likelihood, in which the remaining models are discarded. A more efficient use of the models is to use them, together with the best model, to form a combined learner, using predictions from all models to generate one final prediction. That is, the ensemble of models are used together to form a combined prediction, which hopefully has higher accuracy than the models separately. Ensemble methods are well-known within the field of machine learning and have been shown to produce classifiers with improved predictive performance compared to its individual classifiers [40]. It also allows for some variation in the parameter estimates of the HMM, capturing some of the uncertainty and making the predictions from the model more robust in the initial estimates.

The results from each classifier are combined to produce the ensemble classifier, where the "credibility" of each classifier, i.e. its weight in the ensemble, has to be specified. Without any prior information as to the suitability of each model, it is reasonable to give more weight to models that produce better fit scores. The fit of the HMMs was assessed through ICs, specifically the BIC. The method for calculating posterior model probabilities using the BICs, given in the section on ICs above, can conveniently be used to weigh the results from each classifier in the ensemble. The ensemble classifier then

outputs the class with the largest total weight in the ensemble. Formally, we can express the ensemble classification as follows

$$\hat{C}^E_{t,t+s} = \max_{\hat{C}} \sum_{i=1}^{n} \omega_i \cdot \mathbb{I}_{[\hat{C}]}(\hat{C}^i_{t,t+s}), \tag{5.14}$$

where ω_i are weights, defined in equation 5.7, $\hat{C} \in \{1, 0, 1\}$, n is the number of single classifiers and $\hat{C}^i_{t,t+s}$ are their corresponding predictions.

The classification accuracy of the ensemble will be evaluated by training 6 HMMs on a time window of data (1 hour of observations), generating observations from the predictive distributions and then finding the true and the estimated class, for each classifier. They are then combined to form the ensemble classifier, after which a prediction is produced. The time window is then moved 1 minute ahead and the process is repeated. The accuracy is recorded for $s = 15, 30, 45, 60, 75, 90, 120, 180s$ in equation 5.10, corresponding to predictions of the trend for different horizons. The results can conveniently be summarized in confusion matrices, which display the distribution of the predicted classes for each true class.

Chapter 6

Results

6.1 Training

The implemented algorithm was first trained on simulated data using the simplest ZIP-HMM with 2 states and 2 mixture components (1 Poisson and 1 Dirac). Parameter trajectories $(\hat{\lambda}_1(t), \hat{\lambda}_2(t))$ of the estimated Poisson parameters as functions of the length of the training sequence are plotted in figure 6.1, together with the true parameters of the simulation.

The BW algorithm appears to be able to locate the true values of the parameters, indicated by the parameter trajectories converging to the true values of the Poisson components. The results suggest that the convergence properties of the algorithm are sensitive to the initial estimates used, even in this toy example where the generating model is itself a HMM. Two different initial estimates may yield similar results but one may require many more iterations than the other before converging. This implies that proper initialization is highly important when training HMMs.

The BW algorithm, using the HMM toolbox in [32], was ran on the prototype HMM with Gaussian mixtures as emission distribution can be found in figure 9.2 in the appendix, showing parameter trajectories for the mixture distribution in a GM-HMM with 2 states and 2 mixture components. Again, the EM-algorithm appear to move close to the true value of the generating distribution. This indicates that the implementation of the EM-algorithm in this thesis does not behave badly compared to standard results for HMMs.

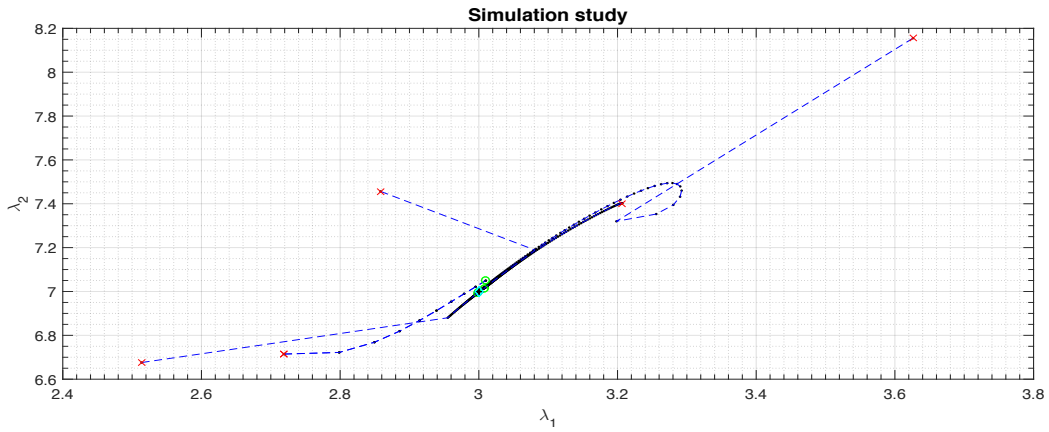

Figure 6.1: *Plot of 5 parameter trajectories for the simulated data, using a ZIP(2,2). The x-axis shows the Poisson parameter in first state and the y-axis shows the parameter in the second state. The red x:s marks the initial guesses for each run of the EM-algorithm, and the green circles show the final values. The true value for the simulated data is indicated by the cyan diamond. The sequence length was set to be 10 000 and the algorithm was allowed to extensively search the parameter space by setting the maximum iterations allowed and convergence threshold to be 600 and* 10^{-8}*, respectively.*

The model distance, defined in equation 5.4 above, was also calculated for the estimated ZIP-HMM model and the results can be found in figure 6.2 below. The figure shows the distance, between the best of 10 estimated models, to the true model as a function of the length of the training sequence. The distance decreases with increasing sequence length, which is expected since the parameters of the estimated model converge at some point, after which additional data has little effect on the parameter values. The rate of convergence of the distance to its limit appears to rapidly decrease after about 4 000 observations, and the distance does not decrease much with increasing sequence length after this point. A sequence length of 90 000 observations (not shown in the plot) produced a distance of 0.0101, which is only slightly smaller than the final value in the plot. This does not necessarily imply that this is the limit of the distance. Instead, it implies that there is little to be gained, in terms of finding the true model, when using more than approximately 7 000 observations. In fact, the distance is within 1% of its value at 90 000 observation, after approximately 4 000 observations, which implies that this is the minimum number of observations that should be used when training the model on real data.

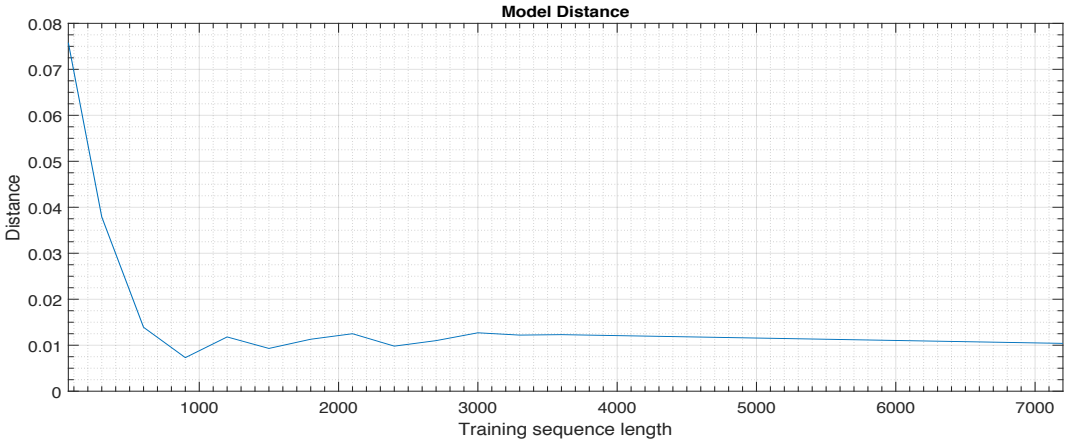

Figure 6.2: *Plot of the model distance as a function of the length of the training sequence, for the ZIP(2,2) model trained on simulated data from a ZIP(2,2) model.*

It might be somewhat surprising that the model distance does not approach zero when the estimated Poisson parameters converge to their true value. The explanation for this is that Poisson parameters only constitute a subset of the parameters of the HMM. That is, the remaining estimates (for the transition matrix, initial distribution and the mixture weights) are not equal to their true values, yielding two different HMMs, and consequently the distance is not zero. In fact, the limit result in equation 5.3 suggest that the distance only becomes zero if the two HMMs are indistinguishable.

6.2 Fit

The results from training the HMMs on the full training data set (08:00-16:00) for different dimensions can be found in table 6.1. The BIC appears to favour, with some margin, the simplest model possible, that is the ZIP(2,2) model. The conclusion from the AIC is similar to the BIC with the ZIP(2,2) model being favoured, except that the AIC is also in favour of the ZIP(2,5) model. This model has the largest log-likelihood value of all models but requires 24 parameters to be estimated, compared to the 12 for the ZIP(2,2). As mentioned earlier, the BIC is the preferred IC, therefore the ZIP(2,2) model is the one chosen.

The large variations in the number of iterations, and consequently run-time, made in the different runs again demonstrate the sensitivity of the EM-algorithm to the initial parameter estimates. In particular, the ZIP(2,5) model arrived at the largest likelihood in less than half of the iterations made in the best ZIP(2,2) run.

K	D	Log-likelihood	BIC	AIC	Iterations	Run-time [s]
2	2	-11730,606396	23584,430360	23485,212792	16	80,484
3	2	-11730,990380	23677,611504	23503,980760	15	83,875
2	4	-11736,356553	23678,075719	23512,713106	23	124,406
2	5	-11718,575588	23683,586312	23485,151176	6	27,781
3	3	-11734,663077	23746,565682	23523,326154	34	207,969
4	2	-11726,517317	23781,614815	23517,034634	21	105,219
2	3	-11809,344191	23782,978473	23650,688382	44	246,484
3	4	-11761,327526	23861,503364	23588,655052	135	890,063
5	2	-11734,433567	23930,933014	23558,867134	23	116,047
3	5	-11761,327526	23923,112148	23600,655052	135	890,063

Table 6.1: *Dimensions of the 10 best models, with respect to the BIC, for the EURUSD all trained using the data from 08:00 to 16:00. Each dimension shows the best model, i.e. largest log-likelihood over 10 runs, The table also shows the AIC, number of iterations made and the total run-time of the algorithm (all algorithm runs were performed using the same MATLAB settings and computer, making them comparable)*

Learning curves for the best model are given in table 6.1 and figure 6.3. The curves show that the emission distribution parameter estimates, in both states, have essentially converged after 60 minutes of training data. Adding more training data has little to no effect on the parameter values, suggesting there is a diminishing return in information content, for the model, of the additional data. This can be compared to the study of the model distance described earlier, where the distance had converged for the training sequence of about 3500 observations, which is almost exactly one hour of data.

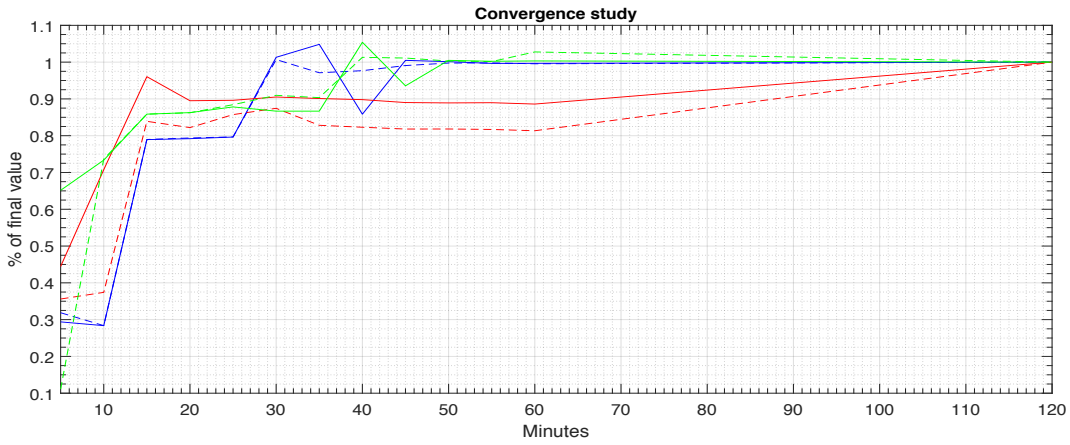

Figure 6.3: *Parameter values for the mixture distribution as functions of the number of training data points, for the EURSEK with $K = 2, D = 2$. The solid lines are for the Poisson parameter in the state with the largest weight component for the Dirac, and the dotted lines represent the Poisson parameter in the other state. The colors represent the data sequence used, with blue,red and green corresponding to training sequences beginning at 08:00, 12:00 and 14:00. The other sequences showed similar results.*

Using the results from the Learning Curve experiments, shorter models using 1-hour long training sequences were estimated for each of the dimensions listed in table 6.1, for each hour in the interval 08:00-16:00. Some of the results can be found in table 6.2. The results shows that the ZIP(2,2) model is preferred in all of the cases, with the general results being similar to those obtained for the larger models in table 6.1. Hence, the ZIP(2,2) model will be used in the following analysis.

08:00 - 09:00						
K	**D**	**Log-likelihood**	**Run-time [s]**	**Iterations**	**BIC**	**AIC**
2	2	-2678,718161	15,156	19	5400,111952	5381,436322
3	2	-2672,345761	41,828	51	5419,373875	5386,691523
2	4	-2689,782895	25,781	35	5450,691840	5419,565790
12:00 - 13:00						
K	**D**	**Log-likelihood**	**Run-time [s]**	**Iterations**	**BIC**	**AIC**
2	2	-1513,094183	16,510	21	3068,863997	3050,188367
3	2	-1520,202850	15,641	18	3115,088052	3082,405699
2	4	-1515,591067	11,688	15	3102,308183	3071,182133
15:00 - 16:00						
K	**D**	**Log-likelihood**	**Run-time [s]**	**Iterations**	**BIC**	**AIC**
2	2	-1553,600253	8,594	11	3149,876136	3131,200506
3	2	-1548,859225	7,078	9	3172,400802	3139,718449
2	4	-1566,217777	36,453	50	3203,561604	3172,435554

Table 6.2: *Table of results from the EM-algorithm ran on models, trained using 1 hour of data. Models of all dimensions in table 6.1 were analyzed and the 3 best during each time periods are presented here.*

6.3 Performance

Plots of the predictions from the HMM and the GBM can be found in figure 6.4, for data trained during 08:00-09:00, calculated at the values given in the header of table 6.3. The mean and the standard deviation for the predictions quickly shows little variation for the HMM, indicating that the underlying MC has converged, after which the predictive distribution does not change with time. The GBM on the other hand shows the behaviour expected according to equation 2.6, with the mean being close to S_0, or the last observation in data, due to the small value for the trend $\mu \approx -3.5 \cdot 10^{-7}$. The variance shows a more rapid increase, due to the larger value for the σ parameter ($\approx 5.3 \cdot 10^{-5}$), and the normally distributed variance for the Brownian motion driving the GBM. This can also be noted from the resemblance of the standard deviation estimates for the GBM in figure 6.4, to a sideways Gaussian "bell". The prediction mean for the HMM is, however, not zero but instead it's determined by the predictive distribution. In the figure, the prediction mean is slightly below the last observation.

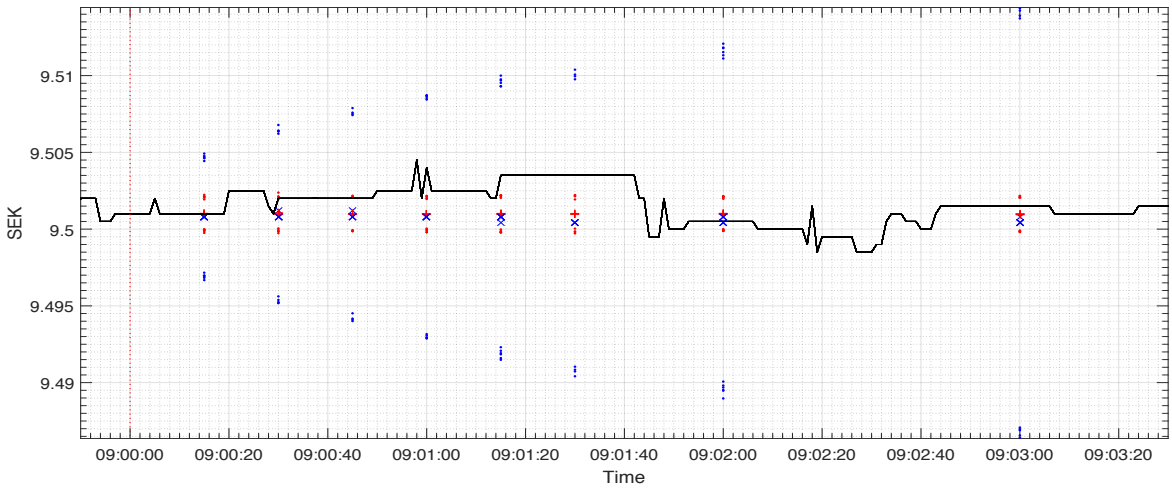

Figure 6.4: *Plot of predictions for the price using the HMM(red) and the GBM(blue). The black line shows the true price process. The red "+" shows the prediction means and the red dots shows 2 times the standard deviation in the predictions, generated using 1000 draws, expressed in pips. The blue crosses and dots show the means and bounds for the GBM. The red dotted vertical line to the left shows the last observation used in the training data. As noted earlier, the bounds for the HMM are essentially identical, after approximately 60 seconds, indicating that the chain converged to the stationary distribution.*

A similar study as the one in figure 6.4 for every trading hour of the day can be found in tables 6.3 and 6.4, with the difference that the MPE and SDPE are studied instead of the mean and the standard deviation of the predictions. The table show that the behaviour of the HMM and GBM are similar for the different times of the day. In particular, the error in the predictions appears to closely follow the trading intensity, as given in figure 3.3 of the average turnover. Larger turnover implies higher market activity, which in turn implies a larger variation in the price process. The errors, for both the HMM and the GBM, follow the U-shape in the turnover curve, and taking its minimal value in the bottom of the curve.

The HMM does appear to perform better than the GBM, as the MPE is smaller for the HMM for almost all prediction hours and horizons with some exceptions in table 6.4 for longer horizons. It is worth to note, however, that due to the small variance in the predictions for the HMM, the true value is often not within the $\pm$ bounds stated, whenever the magnitude of the MPE is larger than approximately 5 pips. This is not the case for the GBM, which almost always captures the true value within the bounds, due to their larger sizes.

Overall, the performance of the HMM is comparable to the GBM. The HMM often generates smaller MPEs but the GBM generates larger confidence bounds containing the true value. In general, the performance becomes worse, with respect to the MPE, for

Time	15 s	30 s	45 s	60 s
08:00-09:00	0(1)±5(19)	-10(-11)±5(27)	-10(-11)±6(34)	-15(-18)±5(39)
09:00-10:00	-5(-10)±5(12)	0(2)±4(17)	15(16)±4(21)	15(17)±4(25)
10:00-11:00	0(0)±2(7)	0(-1)±3(10)	0(-1)±2(12)	0(-2)±2(14)
11:00-12:00	0(0)±2(7)	0(0)±2(10)	10(10)±2(12)	10(9)±2(13)
12:00-13:00	0(2)±4(11)	0(1)±4(16)	-10(-8)±3(19)	0(3)±4(23)
13:00-14:00	0(-1)±4(8)	0(0)±3(12)	0(-2)±4(15)	0(-1)±4(16)
14:00-15:00	-10(-10)±4(9)	-10(-10)±3(14)	5(6)±4(16)	5(5)±4(20)
15:00-16:00	0(0)±4(11)	10(9)±4(15)	5(4)±4(18)	10(9)±3(22)

Table 6.3: *Prediction accuracy, measured as described in the method section (equation 5.8), for the HMM and the GBM, for different prediction horizons and different times during the day. The entries show the MPE ± SDPE for the HMM, with the corresponding values for the GBM given in the parentheses. The values were calculated using 1000 draws.*

both models with increasing prediction horizons, which can be noted by comparing tables 6.3 and 6.4. It is, however, not evident from these tables when the predictions of the HMM are no longer informative. This issue can conveniently be assessed, quantitatively, in the classification setting.

Time	75 s	90 s	120 s	180 s
08:00-09:00	-25(-28)±6(42)	-25(-29)±6(48)	5(1)±6(55)	-5(-10)±5(66)
09:00-10:00	15(18)±4(27)	40(42)±4(28)	40(44)±4(34)	25(31)±4(42)
10:00-11:00	0(-2)±1(16)	0(-3)±3(18)	0(-3)±2(21)	0(-5)±1(26)
11:00-12:00	10(9)±1(15)	15(15)±2(17)	25(24)±2(19)	0(-2)±2(23)
12:00-13:00	-5(-2)±3(25)	0(5)±4(28)	-20(-17)±4(33)	10(18)±4(39)
13:00-14:00	20(17)±3(18)	20(17)±3(21)	20(17)±3(24)	20(15)±4(29)
14:00-15:00	5(5)±4(21)	5(5)±4(23)	5(6)±4(27)	-80(-80)±3(33)
15:00-16:00	20(20)±4(24)	25(24)±4(26)	5(2)±4(31)	10(6)±4(39)

Table 6.4: *Prediction accuracy, same as in table 6.3, with longer prediction horizons.*

Results from the trend prediction can be found in table 6.6, showing confusion matrices for the different prediction horizons. These were calculated using sliding data windows and 6 HMMs, chosen to balance computational load and accuracy, for each window to produce the ensemble classifier. A probabilistic component was introduced to the classifier defined in equation 5.13, by assigning probabilities to each class and then

sampling from the resulting distribution. That is, probabilities were calculated for the distances in equation 5.12 using an exponential distribution. These 3 probabilities were then normalized to form a discrete distribution, from which a class was drawn. This way, the closest distances has the highest probability of being chosen, but with the addition of uncertainty to the classifier through the probabilities. The λ parameter for the exponential was found by calculating the distances from the points, corresponding to the true class, to the last observation. Fifty simulations using this method were ran for each horizons and the average of the respective confusion matrices is shown in table 6.6.

The table shows that the sensitivity of the ensemble classifier is essentially that of a random classifier for which all entries are equal to $1/|\mathscr{C}|$, where $\mathscr{C}$ is the set of classes. This implies that the classifier has no apparent discriminative advantage over a random classifier as far as detecting the true trend. In other words, the performance of the classifier is not worse compared to that of a random classifier.

	-1	**0**	**1**
15 s	20(6)%	45(5)%	1(12)%
30 s	7(7)%	37(13)%	2(5)%
45 s	-5(7)%	21(15)%	1(6)%
60 s	-17(7)%	-14(11)%	3(6)%
75 s	-12(3)%	-23(7)%	-4(6)%
90 s	-18(6)%	-44(10)%	-4(6)%
120 s	-19(6)%	-50(12)%	-4(5)%
180 s	-23(5)%	-80(7)%	-6(6)%

Table 6.5: *Values of the F_β-measure, with $\beta = 1/2$, calculated for the confusion matrices in table 6.6.*

The sensitivity of the classifier affects its precision. Specifically, the precision will depend on how many counts there are of each true trend, as these become evenly distributed across the class predictions. Hence, the precision, as defined in equation 2.1 is somewhat unreliable. For the other definition of precision, given in equation 2.3, the ensemble classifier shows better performance compared to a random classifier.

Table 6.6 does not show the variation in the classifier, which instead can be found in table 6.5. This table shows the relative difference in the F_β-measure (equation 2.2) for the ensemble classifier compared to that of a random classifier, calculated from 50 simulations of the confusion matrices, together with the standard deviation of the values. From this table it's evident that the performance of the price model is consistently bad, compared to the random generator, for prediction horizons longer than 45 seconds. Reasonably,

the best performance is obtained for the shortest prediction horizon.

(a) Prediction horizon: 15 s

	-1	**0**	**1**
-1	41	38	40
0	84	86	92
1	13	14	15

(b) Prediction horizon: 30 s

	-1	**0**	**1**
-1	41	40	45
0	56	55	60
1	41	43	42

(c) Prediction horizon: 45 s

	-1	**0**	**1**
-1	41	39	42
0	47	47	46
1	51	53	57

(d) Prediction horizon: 60 s

	-1	**0**	**1**
-1	41	39	42
0	30	31	30
1	69	71	70

(e) Prediction horizon: 75 s

	-1	**0**	**1**
-1	50	49	51
0	29	28	27
1	60	64	65

(f) Prediction horizon: 90 s

	-1	**0**	**1**
-1	50	48	52
0	18	19	19
1	77	73	73

(g) Prediction horizon: 120 s

	-1	**0**	**1**
-1	51	51	55
0	17	17	16
1	71	71	75

(h) Prediction horizon: 180 s

	-1	**0**	**1**
-1	54	58	59
0	7	7	7
1	77	76	78

Table 6.6: *Confusion matrices for each prediction horizon, calculated as described in the method section. Each element is the average of 20 runs, such that the total count is preserved. The rows show the true classes and the columns show the predicted classes, such that the entry in row i and column j show the number of class j predictions when the true class is i.*

6.4 Trading

Due to the computational load of the HMM ensemble learner, it was not included as a trend predictor in the strategy. Recalculating and updating the model proved to be too time consuming, although the model showed some promise in the previous results. Furthermore, the modular form of the strategy implies that it can be used without a price model. The placement of limits orders was randomized (which is essentially a random walk for the price process) for each bucket, while verifying that the expected value for the total trading time in each bucket did not exceed the duration of the buckets.

The set parameter values used in the simulations of the trading strategy, for EURSEK, are

$$S^d + S^T = 100, MI = 10\%, TR = 1,$$

The estimation procedure described for obtaining an estimate of the execution rate yielded the value 0.015, which corresponds to an average value of $1/0.015 \approx 70$ seconds. That is, on average, a limit order requires 70 seconds before being filled. If the average size of a limit buy order is 1 million EUR, a bucket where 10 orders are to be placed then requires 700 seconds on average for the orders to be filled. This corresponds to $700/1800 \approx 40\%$ of the total bucket size, or bucket duration.

Using these settings, trading simulations were ran 10-15 different times for the start of the trading, depending on the horizon required for the strategy. For each starting time, 1000 simulations were ran for the trading simulations and the results were recorded. This was repeated for different values of α, and the results can be found in figure 6.5 below. Duration is the time required for the strategy from start to the end of trading. The profit is defined as the difference between the market TWAP and the strategy VWAP, expressed in pips. These values can also be found in table 9.1, together with their corresponding standard deviations.

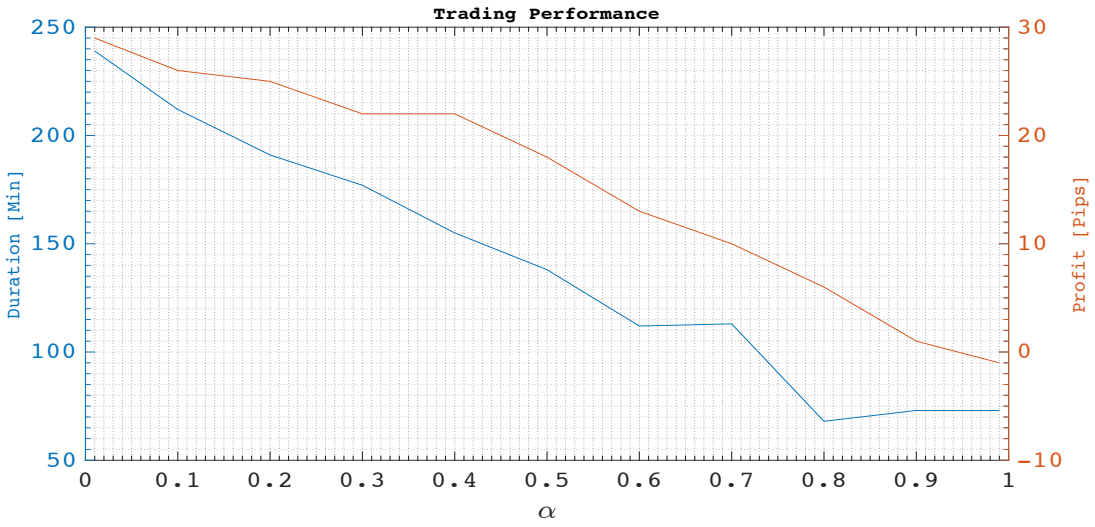

Figure 6.5: *Trading performance for the strategy as a function of the risk aversion parameter* α.

As expected, the strategy generates larger profits when the VWAP part of the strategy accounts for a larger share of the total volume. The larger profit comes at the expense of time, as longer trading horizons are required. The profit and duration curves appear to be quite stable over the day, which can also been seen in table 9.1 in the appendix.

A plot of the cumulative volume distribution, as a function of time, for several simulations can be found in figure 6.6 in the appendix. They demonstrate the variation in the trading duration due to the probabilistic nature of the limit orders.

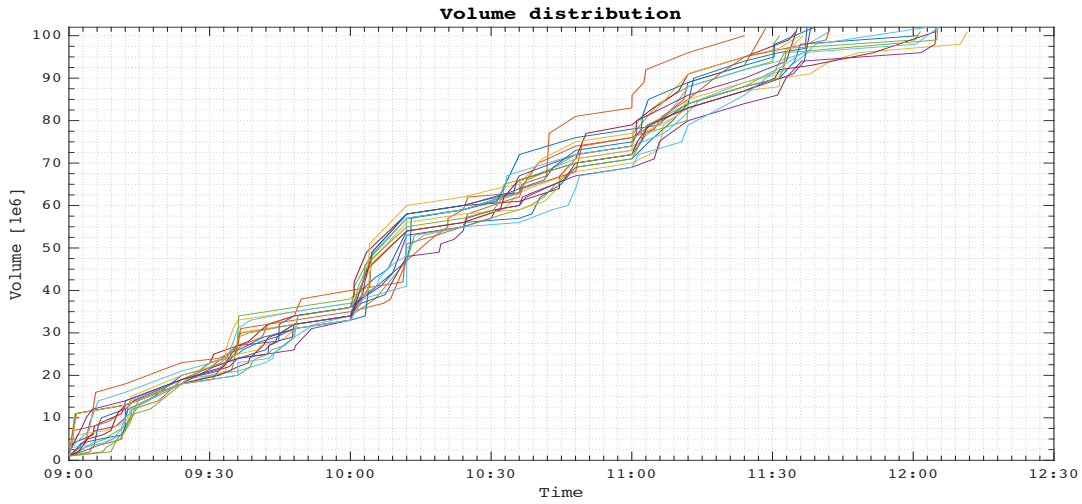

Figure 6.6: *Plot of the cumulative volume distribution over time, for trading started at* $09:00$ *with* $\alpha = 0.4$. *Note that a TWAP algorithm would produce a line with slope 1, while a VWAP can produce curves of many different forms.*

One curious result is the distribution of the profits for the different simulations. Figure 6.7 shows a histogram of the profit distribution, with trading starting at $09:00$, $\alpha = 0.4$, and 1000 simulations. The distribution is clearly multimodal, with two local maxima (one around 15 pips and the other around 30 pips). This behaviour was observed for the majority of the trading simulations and also in the trading durations. Another noteworthy aspect of the histogram is the small number of simulations yielding negative profits, indicating that the strategy is performing well compared to the benchmark, the TWAP.

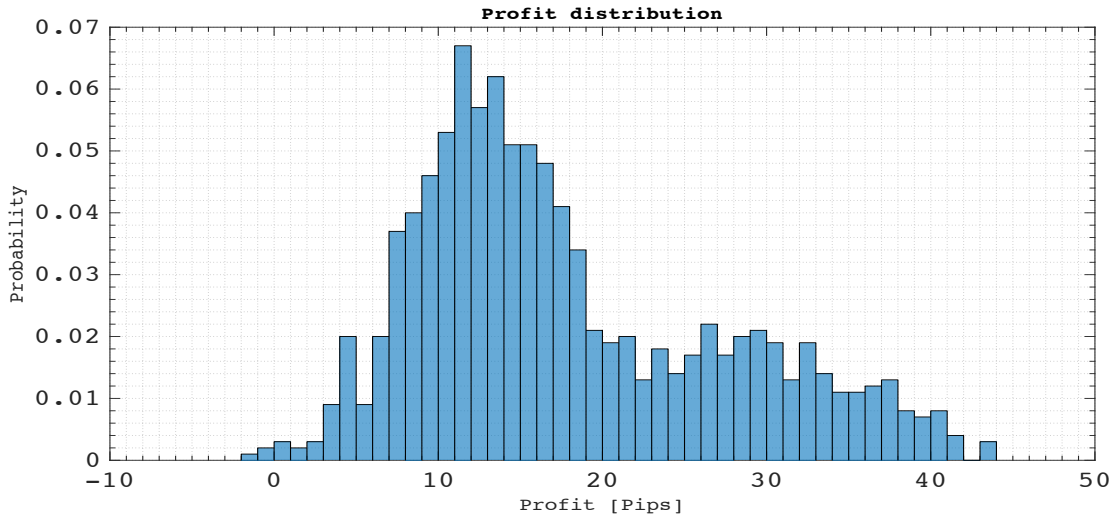

Figure 6.7: *Histogram of the profit distribution for trading started at* $09:00$ *with* $\alpha = 0.4$*, estimated using 1000 simulations of the trading strategy.*

Chapter 7

Discussion

This section provides an analysis and discussion of the results, with the disposition being split into two parts: one for the price model and one for the strategy. In short, the HMM, individually for price predictions and as an ensemble for classification, outperforms a random walk for some prediction horizons. The simplest possible version of the strategy, without a price model driving execution decisions, appears to perform well on the data, although more testing is required before any general conclusions can be made.

7.1 ZIP-HMMs

The initial idea of the project was to use HMMs for modelling high-frequency exchange rates. Poisson distributions were used as the price data obtained is discrete, but other discrete distributions can be used in the modelling. The introduction of ZIP models was to provide the HMMs with some flexibility to accommodate for the large amount of zeros in the data. Indeed, roughly a third of the observations generated zeros. This increased flexibility. In comparison, HMMs with emission distributions given by Gaussian mixtures were also estimated for the data (results not shown here). These estimation algorithms generally ran into trouble, in part due to the excess amount of zeros, but also due to the discrete nature of the data. The ZIP models proved to be more stable as well as more accurate, providing credibility and justification to the use of discrete HMMs when modelling FX price data. The estimation algorithm also appear to be more stable for discrete models, which can be seen from the studies performed on simulated data.

As to the performance of the ZIP-HMMS, in both the price and trend prediction, it outperforms a random walk in some cases. Specifically, the HMM is superior for shorter prediction horizons, after which the performance becomes worse. This is an expected property of the HMM, due to convergence of the hidden MC to the stationary distribution. That is, the HMM "forgets" it's filter and becomes independent of time, which is not desirable for a predictive model. This is, however, not necessarily a drawback for

HMMs in particular but for predictive models in general. One solution is to update the HMMs by re-estimating the parameters, but this can quickly become computationally intensive. The usefulness of the HMM, or any price model, should therefore be measured as a trade-off between the predictive performance and the computational intensity.

One drawback of the HMM, however, is the implicit geometric state distributions. That is, the state transitions are governed by the discrete MC, for which the probability of staying in one state has a geometric distribution, with the parameter given by the self-transition probability. This is a limitation of the HMM and there is no general reason for why this is a good approximation for the price process, other than providing an approximation of the actual state durations. It is possible to extend the duration distributions of HMMs to any arbitrary distribution by using so-called Hidden Semi-Markov models. This was out of the scope for this thesis and was not further investigated.

A somewhat surprising result is the small size of the HMMs preferred by the information criterion. Specifically, the ZIP(2,2) was the dominant model in all tests, outperforming other models with a margin. This naturally raises the question of what the states are actually detecting. Looking at the weights of the mixture distributions in each state, the HMMs appear to overwhelmingly favour zeros in one state and a more even distribution between the mixture components in the other state. Indeed, the weight for the Dirac component was observed to be close to 1 in some cases. The interpretation of these results is that the price process appears to have two phases: one in which the price is inactive, making few jumps and shows little movement, and one in which the price is active and can move in both directions. It is possible that more information about the two phases of the price process can be obtained by studying the order book, which is intimately related to the price process. Order book data could then be incorporated into the HMM, which is univariate in this thesis, which could lead to a better understanding of the real-world phenomena responsible for the different phases of the price process.

As a final note, the results from the ensemble learner suggest that the HMM could be used to improve the trading performance of the devised strategy. This is mainly due to the ZIP-HMMs ability to deal with the large amount of zeros in the data. This is also the rationale for the alternative definition of precision given in the theory section. That is, the ensemble learner has superior sensitivity for detecting when the true class is zero (i.e. no predicted trend). Together with estimated precision, this provides some justification for the use of HMM to form an ensemble learner.

7.2 Strategy Framework

The proposed strategy framework is fairly general and can be greatly customized to suit the problem at hand. It rests on two simple ideas. The first is that the performance of the trading can be improved by combining the strengths of both the TWAP and the VWAP trading algorithms. The second is that locally optimal decisions are sufficient for

global performance. Maximizing profit over the full trading day requires predictions over the whole day, which is extremely difficult and somewhat of the holy grail of trading. In comparison, predicting trends one hour ahead in FX corresponds to predicting stock prices 10 years ahead. Furthermore, the objective of the trading is unloading a large volume, not proprietary trading. As such, the goal is to trade at the best price possible during the fixed trading horizon, not simply wait for the best price over the day.

The specifications and parameters included in the strategy allows for dynamic trading, able to react to current market conditions, as well as allowing the strategy to be adjusted to client or trader preferences. Deriving theoretical limits on performance, using assumptions for the volume and price processes, is possible using methods from the theory of optimal control. This was, however, out of the scope for the thesis and simulations were instead used to provide some justification as to the performance of the strategy.

One possible criticism against the strategy is that trading on both sides of the spread can adversely affect the market. This is certainly the case, but this not an artifact of the trading, rather it's a consequence of market impact. This limits the volume that can be traded within a time period. Market impact was included in the modelling by setting a bound on the allowed volume to trade in a bucket, for both parts of the strategy. The assumption made was that the trading had no market impact by bounding the allowed volume. In reality, any trading has a market impact but it's dependent on the volume traded. The implication of the assumption is then that the market impact is negligible for volumes below the bound, which is necessary for the strategy to make trades.

The simulation experiments for the strategy were performed in a somewhat idealistic setting, although some of the important advantages of the strategy were neutralized. In particular, due to computational constraints, no price model was used in the simulation, and trading times were instead chosen by random over each bucket. Furthermore, some of the dynamic aspects of the strategy were excluded. Volume redistributions between the VWAP and TWAP part, based on the current performance of the strategy, were forbidden. Finally, limit orders were made by observing the volume and price of the best bid at the randomized trade times, without removing it from the order book. In these settings, trading can be simulated without a model for the order book. Also, the order book changes almost every second in some way, meaning that simulations has a very small probability of filling the same order multiple times. Altogether, the simulations were performed using some advantageous conditions but even more unfavorable ones. Despite this, the strategy performed well compared to the benchmark, offering a proof-of-concept. An extensive amount of testing, using much larger data sets and different functional forms for the $d-$function, is required before any general conclusions can be made. However, the results suggest that further research is warranted.

Chapter 8

Concluding Remarks

8.1 Conclusion

The main objective of this thesis was to study the use of ZIP-HMMs on high-frequency foreign exchange data. The conclusion from the study is that this type of model shows some promise, as a price predictor and a trend predictor using ensemble classifiers. Yet, more research about the properties of HF FX markets, as well as the behaviour for ZIP-HMMs, is required before any general conclusion can be made.

The evaluation of the strategy framework was limited in this thesis, mainly due to time and computational constraints. The initial results were positive, indicating that further research and development of the framework is warranted and should be of interest to concerned parties.

8.2 Future Research

It is possible that the performance of the ZIP-HMM can be improved by linking the Poisson parameters to other market data than the price process. Specifically, regression methods can be used to estimate the λ parameters, which could then be incorporated into the HMM framework. This is a specific example of a more general point that the modelling could probably be improved by including more information about the price process through other forms of market data, thus making the model multivariate.

On a more general note, the predictive, not to mention descriptive, performance of a price model can probably be improved by studying the different physical events that can cause changes in the price process. Indeed, the success of predictive models in the natural sciences is rooted in deep understanding of the behaviour of the system under study. Although the behaviour of financial markets, which at the lowest level is controlled

by the behaviour of traders, is probably much more complex, understanding of this behaviour could lead to substantial developments in the field of financial mathematics.

Chapter 9

Appendix

9.1 Derivation of the BW-algorithm parameter estimates

In the ZIP-HMM, the emission distributions are as follows

$$
\begin{aligned}
P_\theta(O_t = o_t|q_t) &= \sum_{d=0}^{D} P_\theta(o_t, m_t = d|q_t) && (9.1)\\
&= \sum_{d=0}^{D} P_\theta(o_t|m_t = d, q_t) P_\theta(m_t|q_t) \\
&= \sum_{d=0}^{D} P_\theta(o_t|m_t = d, q_t) \times w_{dk} \\
&= \mathbb{I}(o_t)_{[0]} \times w_{0k} + \sum_{d=1}^{D} \frac{\lambda_{dk}^{o_j} e^{-\lambda_{dk}}}{o_j!} \times w_{dk}. && (9.2)
\end{aligned}
$$

Baum's auxiliary function is given as follows

$$
Q(\theta, \theta') = \sum_{\bar{q} \in \mathcal{Q}} \sum_{\bar{m} \in \mathcal{M}} \log(P_\theta(\bar{o}, \bar{q}, \bar{m})) P_{\theta'}(\bar{q}, \bar{m}|\bar{o}), \tag{9.3}
$$

where evaluation of the right-hand side consitutes the E-step of the algorithm and the maximizing the Q-function with respect to θ consitutes the M-step of the algorithm. Using the Markov property of the underlying chain and the conditional independence of

the HMM, the complete data likelihood can be written in a more convenient format.

$$\begin{aligned} P(\bar{o}, \bar{q}, \bar{m}|\theta) &= P_\theta(o_{1:t}, q_{0:t}, m_{1:t}) \\ &= P_\theta(o_t, m_t|o_{1:t-1}, q_{0:t}, m_{1:t-1}) P_\theta(o_{1:t-1}, q_{0:t}, m_{1:t-1}) \\ &= P_\theta(o_t, m_t|q_t) P_\theta(o_{1:t-1}, q_{0:t}, m_{1:t-1}) \\ &= P_\theta(o_t, m_t|q_t) P_\theta(q_t|o_{1:t-1}, q_{0:t-1}, m_{1:t-1}) P_\theta(o_{1:t-1}, q_{0:t-1}, m_{1:t-1}) \\ &= P_\theta(o_t, m_t|q_t) P_\theta(q_t|q_{t-1}) P_\theta(o_{1:t-1}, q_{0:t-1}, m_{1:t-1}). \end{aligned} \tag{9.4}$$

Repeating this procedure for the last term and collecting terms yields the following factorization of the complete data loglikelihood

$$P(\bar{o}, \bar{q}, \bar{m}|\theta) = P_\theta(q_0) \times \prod_{i=1}^{t} P_\theta(q_i|q_{i-1}) \times \prod_{j=1}^{t} P_\theta(o_j|m_j, q_j). \tag{9.5}$$

This can be used in Baum's Q-function above, yielding

$$\sum_{\bar{q}\in\mathcal{Q}} \sum_{\bar{m}\in\mathcal{M}} \left[\log P_\theta(q_0) + \sum_{i=1}^{t} \log P_\theta(q_i|q_{i-1}) + \sum_{j=1}^{t} \log P_\theta(o_j, m_j|q_j) \right] P_{\theta'}(\bar{q}, \bar{m}|\bar{o}). \tag{9.6}$$

The 3 terms in this expression can now be studied separately. Evaluating the expectation of these 3 terms under the smoothing distribution $P_{\theta'}(\bar{q}, \bar{m}|\bar{o})$ is the E-step of the algorithm. Note that only the 3^{rd} term depends on the form of the emission densities. The first term can be rewritten by marginalizing out variables as follows

$$\begin{aligned} \sum_{\bar{q}\in\mathcal{Q}} \sum_{\bar{m}\in\mathcal{M}} \log P_\theta(q_0) P_{\theta'}(\bar{q}, \bar{m}|\bar{o}) &= \sum_{\bar{q}\in\mathcal{Q}} \log P_\theta(q_0) \sum_{\bar{m}\in\mathcal{M}} P_{\theta'}(\bar{q}, \bar{m}|\bar{o}) \\ &= \sum_{k=1}^{K} \log P_\theta(q_0 = k) P_{\theta'}(q_0 = k|\bar{o}) \\ &= \sum_{k=1}^{K} \log \pi_k \times P_{\theta'}(q_0 = k|\bar{o}). \end{aligned}$$

In the E-step of the algorithm, the second factor in the product above can be evaluated efficientely using the Forward-Backward algorithm. For now, we introduce the notation $\gamma_t(k) := P(q_t = k|\bar{o}, \theta)$ and maximize this expression with respect to π_k, together with the Lagrange constraint $\sum_{k=1}^{K} \pi_k = 1$, which consitutes the M-step of the algorithm.

$$\frac{\partial}{\partial \pi_k} \left(\sum_{s=1}^{K} \gamma_0(s) \log \pi_s + \eta \big(\sum_{j=1}^{K} \pi_j - 1 \big) \right) = 0, \quad \forall k = 1, \ldots, K. \tag{9.7}$$

Solving for each k yields identitcal equations of the form $\gamma_0(k) = -\eta\pi_k$. Summing this equation over $k = 1, \ldots, K$ on both sides and eliminating the Lagrange variable η then yields

$$\pi_k = \frac{\gamma_0(k)}{\sum_{s=1}^{K} \gamma_0(s)}. \tag{9.8}$$

This concludes the M-step for the first term in Baum's Q-function. Using the same reasoning as for the first term, the expressions for the second term can be simplified by marginalizing out variables as follows

$$\begin{aligned}\sum_{\bar{q}\in\mathcal{Q}} \sum_{\bar{m}\in\mathcal{M}} \sum_{i=1}^{t} \log P_\theta(q_i|q_{i-1}) P_{\theta'}(\bar{q}, \bar{m}|\bar{o}) &= \sum_{\bar{q}\in\mathcal{Q}} \sum_{i=1}^{t} \log P_\theta(q_i|q_{i-1}) \sum_{\bar{m}\in\mathcal{M}} P_{\theta'}(\bar{q}, \bar{m}|\bar{o}) \\ &= \sum_{\bar{q}\in\mathcal{Q}} \sum_{i=1}^{t} \log P_\theta(q_i|q_{i-1}) P_{\theta'}(\bar{q}|\bar{o}). \end{aligned} \tag{9.9}$$

Marginalizing out variables and introducing the short-hand notation $\xi_t(i,j) = P_{\theta'}(q_{t-1} = i, q_t = j|\bar{o})$ and $a_{ij} = P_\theta(q_i = r|q_{i-1} = s)$ yields

$$\sum_{\bar{q}\in\mathcal{Q}} \sum_{i=1}^{t} \log P_\theta(q_i|q_{i-1}) P_{\theta'}(\bar{q}|\bar{o}) = \sum_{i=1}^{t} \sum_{r=1}^{K} \sum_{s=1}^{K} \xi_i(s,r) \log a_{sr}. \tag{9.10}$$

As for the first term above, $\xi_t(i,j)$ can be evaluated effiecently using the Forward-Backward algorithm. Maximizing this last expression with respect to a_{sr} consitutes the M-step and the calculations are similar to the ones for the first term.

$$\frac{\partial}{\partial a_{sr}} \Big(\sum_{i=1}^{t} \sum_{r=1}^{K} \sum_{s=1}^{K} \xi_i(s,r) \log a_{sr} + \eta \big(\sum_{j=1}^{K} a_{sj} - 1 \big) \Big). \tag{9.11}$$

Again, taking the derivative $\forall s, r = 1, \ldots, K$ yields

$$\sum_{i=1}^{t} \xi_i(s,r) = -\eta a_{sr}. \tag{9.12}$$

Using the same method as above for eliminating the Lagrange variable yields

$$a_{sr} = \frac{\sum_{i=1}^{t} \xi_i(s,r)}{\sum_{j=1}^{K} \sum_{i=1}^{t} \xi_i(s,j)}. \tag{9.13}$$

The first and second term in Baum's Q-function do not depend on the form of the emission distributions and are therefore always have the form given in the equations above. The third term, however, does depend on the form of the emission distribution

and consequentely, so does both the E-step and the M-step for it.

$$\sum_{\bar{q}\in\mathcal{Q}}\sum_{\bar{m}\in\mathcal{M}}\sum_{j=1}^{t}\log P_\theta(o_j,m_j|q_j)P_{\theta'}(\bar{q},\bar{m}|\bar{o}) = \sum_{j=1}^{t}\sum_{k=1}^{K}\sum_{d=0}^{D}\log P_\theta(o_j,m_j=d|q_j=k)P_{\theta'}(q_j=k,m_j=d|\bar{o}). \quad (9.14)$$

Separating the degenerate component from the Poissons yields

$$\sum_{j=1}^{t}\sum_{k=1}^{K}\Bigg[\log P_\theta(o_j,m_j=0|q_j=k)\times P_{\theta'}(q_j=k,m_j=0|\bar{o}) + \sum_{d=1}^{D}\log P_\theta(o_j,m_j=d|q_j=k)\times P_{\theta'}(q_j=k,m_j=d|\bar{o})\Bigg]. \quad (9.15)$$

Using the definition of conditional probability to rewrite $P_\theta(o_j,m_j=d|q_j=k)$ as $P_\theta(o_j|m_j=d,q_j=k)P_\theta(m_j=d|q_j=k)$ and introducing the notation $w_{dk}:=P_\theta(m_j=d|q_j=k)$ yields, in the expression above,

$$\sum_{j=1}^{t}\sum_{k=1}^{K}\Bigg[\log(w_{0k})P_{\theta'}(m_j=0,q_j=k|\bar{o}) + \sum_{d=1}^{D}\log(\frac{\lambda^{o_j}e^{-\lambda_{dk}}}{o_j!}w_{dk})P_{\theta'}(m_j=d,q_j=k|\bar{o})\Bigg]. \quad (9.16)$$

Completing the E-step requires evaluating the smoothing distribution ($\bar{o}$ denotes all observations, i.e. it is the same as $o_{1:T}$) $P_{\theta'}(m_j,q_j|\bar{o}) = P_{\theta'}(m_j=d|q_j=k,\bar{o})P_{\theta'}(q_j=k|\bar{o})$. We begin by expressing the joint distribution $P_{\theta'}(\bar{o},m_j,q_j)$ in two different ways (in the equations below $o_{\neg j}$ denotes all observations expect the one at time j)

$$\begin{aligned}P_{\theta'}(m_j,q_j,o_j,o_{\neg j}) &= P_{\theta'}(o_j|m_j,q_j,o_{\neg j})P_{\theta'}(m_j,q_j,o_{\neg j})\\ &= P_{\theta'}(o_j|m_j,q_j)P_{\theta'}(m_j|q_j,o_{\neg j})P_{\theta'}(q_j,o_{\neg j}) \quad (9.17)\\ P_{\theta'}(m_j,q_j,o_j,o_{\neg j}) &= P_{\theta'}(m_j|q_j,o_j,o_{\neg j})P_{\theta'}(q_j,o_j,o_{\neg j})\\ &= P_{\theta'}(m_j|q_j,o_j,o_{\neg j})P_{\theta'}(o_j|q_j,o_{\neg j})P_{\theta'}(q_j,o_{\neg j}). \quad (9.18)\end{aligned}$$

Equating these two expressions and solving for $P_{\theta'}(m_j|q_j,o_j,o_{\neg j})$ yields, together with

the conditional independence property of the HMM,

$$\begin{aligned} P_{\theta'}(m_j = d|q_j = k, \bar{o}) &= P_{\theta'}(m_j|q_j, o_j, o_{\neg j}) \\ &= \frac{P_{\theta'}(o_{j|m_j,q_j})P_{\theta'}(m_j|q_j, o_{\neg j})P_{\theta'}(q_j, o_{\neg j})}{P_{\theta'}(o_j|q_j, o_{\neg j})P_{\theta'}(q_j, o_{\neg j})} \\ &= \frac{P_{\theta'}(o_j|m_j, q_j)P_{\theta'}(m_j|q_j)}{P_{\theta'}(o_j|q_j)} \\ &= \frac{P_{\theta'}(o_j|m_j, q_j)P_{\theta'}(m_j|q_j)}{\sum_{d=0}^{D} P_{\theta'}(o_j|m_j = d, q_j)P_{\theta'}(m_j = d|q_j)}. \end{aligned} \tag{9.19}$$

Mulitplying this expression with $P_{\theta'}(q_j = k|\bar{o})$ gives the desired smoothing distribution. The smoothing distribution has a different form for the degenerate component and the Poissons. For the degenerate component, $m_j = 0$, the smoothing distribution is given as follows

$$P_{\theta'}(m_j = 0, q_j = k|\bar{o}) = \begin{cases} 0, & o_j > 0 \\ \frac{w'_{0k}\gamma_j(k)}{w'_{0k}+\sum_{d=1}^{D} w'_{dk}e^{-\lambda'_{dk}}}, & o_j = 0, \end{cases} \tag{9.20}$$

where the $'$ denotes the old parameters. This is because the degenerate component can not generate non-zero observations so the probability is 0 in this case. For the Poisson components, i.e. $d = 1, \ldots, D$, the smoothing distribution is given as follows

$$P_{\theta'}(m_j = d, q_j = k|\bar{o}) = \begin{cases} \frac{w'_{ik}\lambda'^{o_j}_{ik}e^{-\lambda'_{ik}}/o_j!}{\sum_{d=1}^{D} w'_{dk}\lambda'^{o_j}_{dk}e^{-\lambda'_{dk}}/o_j!}\gamma_j(k), & o_j > 0 \\ \frac{w'_{ik}e^{-\lambda_{ik}}}{w'_{0k}+\sum_{d=1}^{D} w'_{dk}e^{-\lambda'_{dk}}}\gamma_j(k), & o_j = 0 \end{cases}, \tag{9.21}$$

Using these two expressions in the third term in Baum's Q-function completes the E-step of the Baum-Welch algorithm and yields the full expression

$$\begin{aligned} &\sum_{\substack{j=1 \\ o_j=0}}^{t}\sum_{k=1}^{K}\Bigg[\log w_{0k}\frac{w'_{0k}}{w'_{0k}+\sum_{d=1}^{D} w'_{dk}e^{-\lambda'_{dk}}}\gamma_j(k) + \\ &\sum_{d=1}^{D}(\log(w_{dk} - \lambda_{dk}))\frac{w'_{dk}e^{-\lambda'_{dk}}}{w'_{0k}+\sum_{d=1}^{D} w'_{dk}e^{-\lambda'_{dk}}}\gamma_j(k)\Bigg] + \\ \sum_{\substack{j=1 \\ o_j>0}}^{t}\sum_{k=1}^{K}\sum_{d=1}^{D}&\big(\log(w_{dk}) + o_j\log\lambda_{dk} - \lambda_{dk} - \log o_j!\big)\frac{w'_{dk}\lambda'^{o_j}_{dk}e^{-\lambda_{dk}}}{\sum_{r=1}^{D} w'_{rk}\lambda'^{o_j}_{rk}e^{-\lambda_{rk}}}\gamma_j(k), \end{aligned} \tag{9.22}$$

To improve readability, the following short-hand notation is introduced

$$
\begin{aligned}
A_j(k) &= \frac{w_{0k}^{'}}{w_{0k}^{'}+\sum_{d=1}^{D} w_{dk}^{'} e^{-\lambda_{dk}^{'}}}\gamma_j(k), \\
B_j(k,d) &= \frac{w_{dk}^{'} e^{-\lambda_{dk}^{'}}}{w_{0k}^{'}+\sum_{d=1}^{D} w_{dk}^{'} e^{-\lambda_{dk}^{'}}}\gamma_j(k), \\
C_j(k,d) &= \frac{w_{dk}^{'} \lambda_{dk}^{o_j} e^{-\lambda_{dk}}}{\sum_{r=1}^{D} w_{rk}^{'} \lambda_{rk}^{o_j} e^{-\lambda_{rk}}}\gamma_j(k),
\end{aligned}
\tag{9.23}
$$

for $k = 1, \ldots, K$ and $d = 1, \ldots, D$, giving the final expression for the E-step

$$
\begin{aligned}
&\sum_{\substack{j=1 \\ o_j=0}}^{t} \sum_{k=1}^{K} \left[\log w_{0k} \cdot A_j(k) + \sum_{d=1}^{D} \big(\log w_{dk} - \lambda_{dk} \big) \cdot B_j(k,d) \right] + \\
&\sum_{\substack{j=1 \\ o_j>0}}^{t} \sum_{k=1}^{K} \sum_{d=1}^{D} \big(\log w_{dk} + o_j \cdot \log \lambda_{dk} - \lambda_{dk} - \log o_j! \big) \cdot C_j(k,d).
\end{aligned}
$$

In the M-step of the algorithm. this expression is maximized with respect to the w_{0k} and the w_{dk}, λ_{dk} for $d = 1, \ldots, D$ and $k = 1, \ldots, K$. We begin with the w_{dk}:s. Together with the Lagrange constraint $\sum_{d=0}^{D} w_{dk} = 1$, taking derivative with respect to w_{dk} yields

$$
\begin{aligned}
\frac{\partial}{\partial w_{dk}} &= \ldots = \\
&= \sum_{\substack{j=1 \\ o_j>0}}^{t} \frac{1}{w_{dk}} \cdot B_j(k,d) + \sum_{\substack{j=1 \\ o_j>0}}^{t} \frac{1}{w_{dk}} \cdot C_j(k,d) + \eta. \qquad (9.24) \\
& \qquad (9.25)
\end{aligned}
$$

Equating this expression to 0 and solving for w_{dk} yields

$$
\sum_{\substack{j=1 \\ o_j>0}} B_j(k,d) + \sum_{\substack{j=1 \\ o_j>0}}^{t} C_j(k,d) = -\eta w_{dk}, \tag{9.26}
$$

for $d = 1, \ldots, D$ and $k = 1, \ldots, K$. Similarily for w_{0k}, we get

$$
\sum_{\substack{j=1 \\ o_j=0}} A_j(k) = -\eta w_{0k}. \tag{9.27}
$$

By combining these two expressions the Lagrange variable η can be eliminated and we

get the following expressions

$$w_{0k} = \frac{\sum_{\substack{j=1 \\ o_j=0}}^{t} A_j(k)}{\sum_{\substack{j=1 \\ o_j=0}}^{t} \left[A_j(k) + \sum_{r=1}^{D} B_j(k,r) \right] + \sum_{r=1}^{D} \sum_{\substack{j=1 \\ o_j>0}}^{t} C_j(k,r)}$$

$$w_{dk} = \frac{\sum_{\substack{j=1 \\ o_j=0}}^{t} B_j(k,d) + \sum_{\substack{j=1 \\ o_j>0}}^{t} C_j(k,d)}{\sum_{\substack{j=1 \\ o_j=0}}^{t} \left[A_j(k) + \sum_{r=1}^{D} B_j(k,r) \right] + \sum_{r=1}^{D} \sum_{\substack{j=1 \\ o_j>0}}^{t} C_j(k,r)}, \; d = 1, \ldots, D, \tag{9.28}$$

Similarly, we can solve for the λ_{dk}:s. (Let's solve for the λ_{dk}:s without the constraint that they should all be larger than zero. It turns out that the inequality is satisfied even without including the constraint.)

$$\frac{\partial}{\partial \lambda_{dk}} = \ldots =$$

$$= \sum_{\substack{j=1 \\ o_j>0}}^{t} -B_j(k,d) + \sum_{\substack{j=1 \\ o_j>0}}^{t} \left(\frac{o_j}{\lambda_{dk}} - 1 \right) C_j(k,d), \tag{9.29}$$

$$\tag{9.30}$$

Equating this to 0 and solving for the λ_{dk}:s yields

$$\lambda_{dk} = \frac{\sum_{\substack{j=1 \\ o_j>0}}^{t} o_j \cdot C_j(k,d)}{\sum_{\substack{j=1 \\ o_j>0}}^{t} B_j(k,d) + \sum_{\substack{j=1 \\ o_j>0}}^{t} C_j(k,d),} \tag{9.31}$$

Note that since all the observations in the nominator are > 0 and $B_j(k,d), C_j(k,d) > 0, \forall (k,d)$, it follows that the λ_{dk}:s satisfy the constraint $\lambda_{dk} > 0$.

9.2 Figures

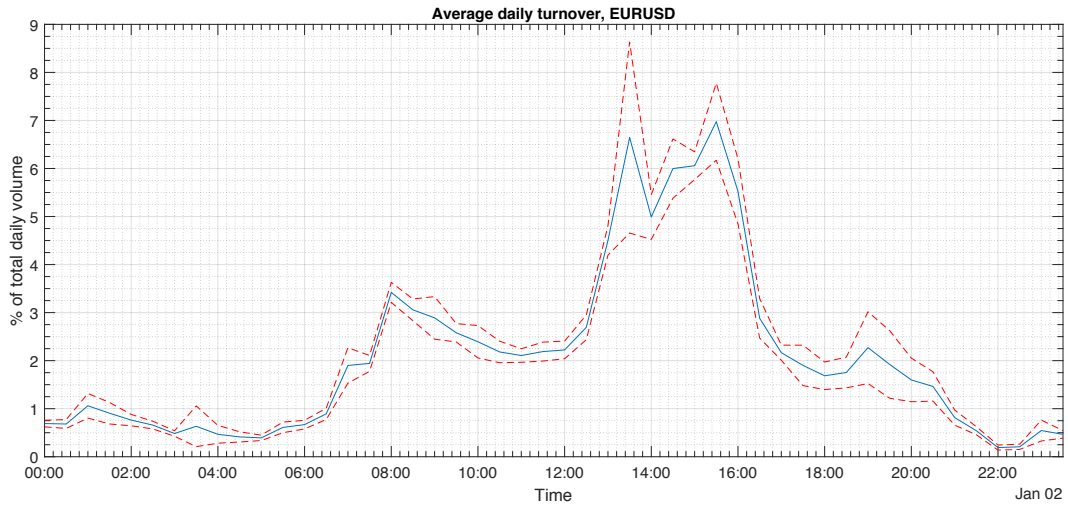

Figure 9.1: *Average turnover for EURUSD.*

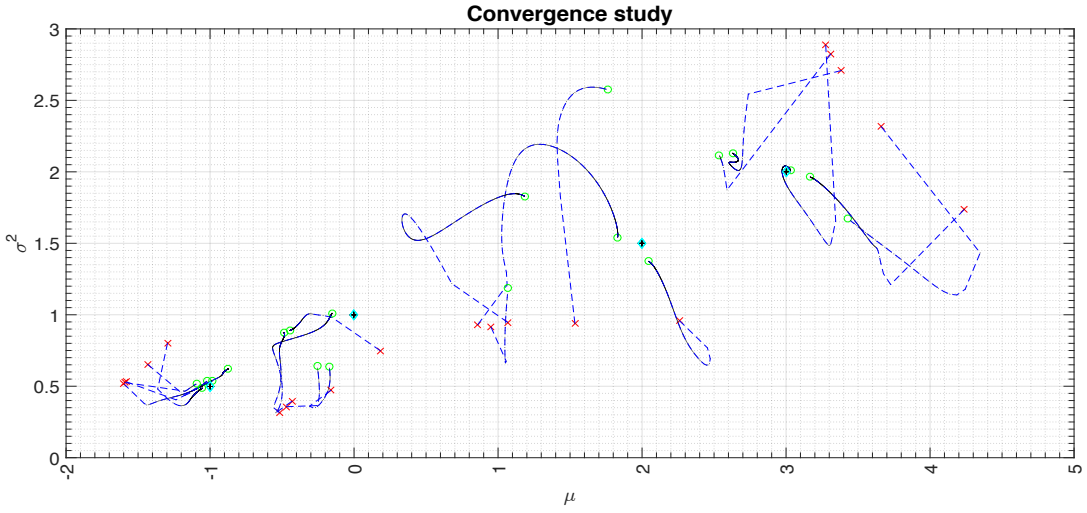

Figure 9.2: *Study of the parameter convergence for the EM-algorithm on simulated data for emission distributions given by Gaussian mixtures, using 10 000 observations. The true values for the mixture components in each state are indicated by the cyan diamond. The blue lines show parameter trajectories as function of iterations. Note that each run of the EM-algorithm produces 4 of the blue lines, each corresponding to one of the 4 Gaussians of the HMM emission distributions.*

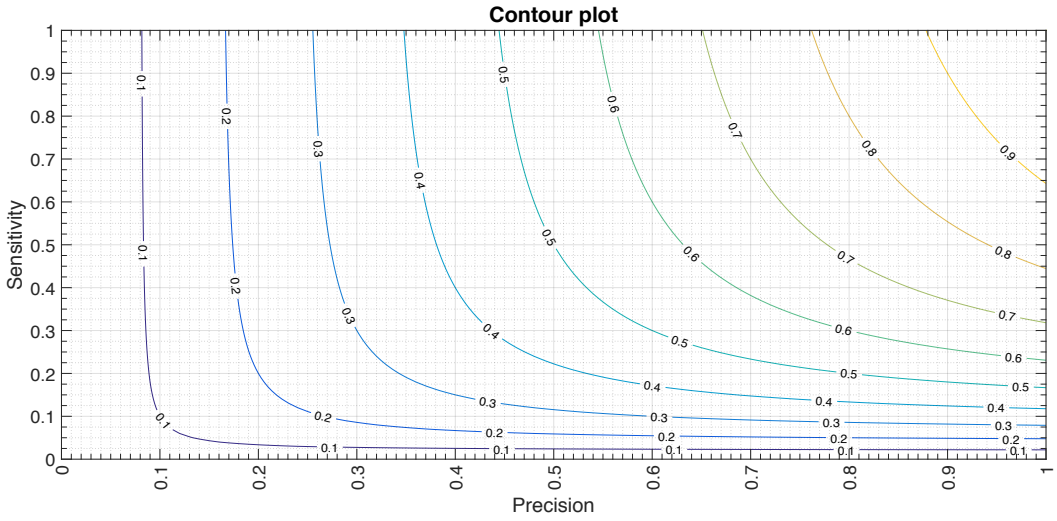

Figure 9.3: *Plot of the F_β-measure for $\beta = 1/2$, with more weight for the precision.*

9.3 Tables

α	**Duration [Min]**	**Profit [Pips]**
0	239(28)	29(16)
0.1	212(8)	26(15)
0.2	191(9)	25(15)
0.3	177(10)	22(13)
0.4	155(15)	22(12)
0.5	138(6)	18(12)
0.6	112(11)	13(11)
0.7	113(10)	10(7)
0.8	68(4)	6(6)
0.9	73(2)	1(4)
1	73(0)	-1(3)

Table 9.1: *Trading performance for the strategy. The table shows the means and standard deviations (in brackets) for the trading duration and the profit, calculated as the difference in pips between the VWAP for the strategy and the market TWAP. The means are plotted in figure 6.5.*

Bibliography

[1] I. Aldridge, *High-frequency trading: a practical guide to algorithmic strategies and trading systems*. John Wiley and Sons, 2009, vol. 459.

[2] R. Almgren and N. Chriss, "Optimal execution of portfolio transactions", *Journal of Risk*, vol. 3, pp. 5–40, 2001.

[3] D. R. Anderson and K. P. Burnham, *Model Selection and Multi-Model Inference: A Practical Information-Theoretic Approach*, 2nd ed. New York: Springer, 2011, ISBN: 978-0-38-722456-5.

[4] R. T. Baillie, T. Bollerslev, and H. O. Mikkelsen, "Fractionally integrated generalized autoregressive conditional heteroskedasticity", *Journal of econometrics*, vol. 74, no. 1, pp. 3–30, 1996.

[5] L. E. Baum and T. Petrie, "Statistical inference for probabilistic functions of finite state markov chains", *The annals of mathematical statistics*, vol. 37, no. 6, pp. 1554–1563, 1966.

[6] J. A. Bilmes, "What hmms can do", *IEICE TRANSACTIONS on Information and Systems*, vol. 89, no. 3, pp. 869–891, 2006.

[7] C. M. Bishop, *Pattern recognition and machine learning*, 1st ed. New York, NY: Springer, 2006, ISBN: 978-0-38-731073-2.

[8] J. Bulla and I. Bulla, "Stylized facts of financial time series and hidden semi-markov models", *Computational Statistics & Data Analysis*, vol. 51, no. 4, pp. 2192–2209, 2006.

[9] O. Cappé, E. Moulines, and T. Rydén, *Inference in Hidden Markov Models*. New York, NY: Springer New York, 2005, ISBN: 978-0-387-40264-2.

[10] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm", *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977, ISSN: 00359246.

[11] S. M. DeSantis and D. Bandyopadhyay, "Hidden markov models for zero-inflated poisson counts with an application to substance use", *Statistics in medicine*, vol. 30, no. 14, pp. 1678–1694, 2011.

[12] R. Durrett, *Essentials of stochastic processes*, 2nd ed. New York: Springer, 2012, ISBN: 978-1-46-143615-7.

[13] A. Dvoretzky, J. Kiefer, and J. Wolfowitz, "Asymptotic minimax character of the sample distribution function and of the classical multinomial estimator", *The Annals of Mathematical Statistics*, pp. 642–669, 1956.

[14] M. R. Hassan and B. Nath, "Stock market forecasting using hidden markov model: A new approach", in *Intelligent Systems Design and Applications, 2005. ISDA'05. Proceedings. 5th International Conference on*, IEEE, 2005, pp. 192–196.

[15] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd. New York, NY: Springer New York, 2009, ISBN: 978-0-387-84857-0.

[16] J. Hull, *Options, futures, and other derivatives*, 8. ed., Global ed.. Boston: Pearson, 2012, ISBN: 978-0-27-375907-2.

[17] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review", *ACM Computing Surveys*, vol. 31, no. 3, 1999.

[18] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning: with Applications in R*, 2nd ed. New York, NY: Springer, 2013, ISBN: 978-1-46-147138-7.

[19] B. Johnson, *Algorithmic Trading and DMA: An introduction to direct access trading strategies*. London: 4Myeloma Press, 2010, ISBN: 978-0-95-639920-5.

[20] B.-H. Juang and L. R. Rabiner, "A probabilistic distance measure for hidden markov models", *AT&T technical journal*, vol. 64, no. 2, pp. 391–408, 1985.

[21] R. Kissell, M. Glantz, and R. Malamut, "A practical framework for estimating transaction costs and developing optimal trading strategies to achieve best execution", *Finance Research Letters*, vol. 1, no. 1, pp. 35–46, 2004.

[22] M. R. Kosorok, *Introduction to Empirical Processes and Semiparametric Inference*, 1st ed. New York, NY: Springer, 2007, ISBN: 978-0-38-774977-8.

[23] D. Lambert, "Zero-inflated poisson regression, with an application to defects in manufacturing", *Technometrics*, vol. 34, no. 1, pp. 1–14, 1992.

[24] I. MacDonald and W. Zucchini, *Hidden Markov and Other Models for Discrete-valued Time Series*, 1st ed. London, UK: Chapman and Hall/CRC, 1997, ISBN: 978-0-41-255850-4.

[25] R. S. Mamon and R. J. Elliott, *Hidden markov models in finance*. Springer, 2007, vol. 4.

[26] P. Massart, "The tight constant in the dvoretzky-kiefer-wolfowitz inequality", *The Annals of Probability*, pp. 1269–1283, 1990.

[27] MATLAB, *version 8.6.0.267246 (R2015b)*. Natick, Massachusetts: The MathWorks Inc., 2010.

[28] G. J. McLachlan and T. Krishnan, *The EM algorithm and extensions*, 2nd ed. Hoboken, N.J.: John Wiley Sons, Inc., 2008, ISBN: 978-0-47-019161-3.

[29] G. J. McLachlan and D. Peel, *Finite Mixture Models*. John Wiley Sons, Inc., 2004, ISBN: 978-0-47-172118-5.

[30] R. A. Meese and K. Rogoff, "Empirical exchange rate models of the seventies: Do they fit out of sample?", *Journal of international economics*, vol. 14, no. 1-2, pp. 3–24, 1983.

[31] S. P. Meyn, *Markov Chains and Stochastic Stability*, 1st ed. Berlin: Springer, 1993, ISBN: 978-0-52-173182-9.

[32] K. P. Murphy, "Hmm toolbox for matlab", *Internet: http://www. cs. ubc. ca/˜ murphyk/Software/HMM/hmm. html,[Oct. 29, 2011]*, 1998.

[33] K. P. Murphy, *Machine learning : a probabilistic perspective*, 1st ed. Cambridge, MA: The MIT Press, 2012, ISBN: 978-0-262-01802-9.

[34] R. M. Neal and G. E. Hinton, "A view of the em algorithm that justifies incremental, sparse, and other variants", in *Learning in Graphical Models*, M. I. Jordan, Ed. Netherlands, NE: Springer Netherlands, 1998, pp. 355–368, ISBN: 978-9-40-115014-9. [Online]. Available: `https://doi.org/10.1007/978-94-011-5014-9_12`.

[35] M. Olteanu and J. Ridgway, "Hidden markov models for time series of counts with excess zeros", eng, *European Symposium on Artificial Neural Networks*, 2012.

[36] ——, "Hidden markov models for time series of counts with excess zeros", in *European Symposium on Artificial Neural Networks*, 2012, pp. 133–138.

[37] T. Petrie, "Probabilistic functions of finite state markov chains", *The Annals of Mathematical Statistics*, vol. 40, no. 1, pp. 97–115, 1969.

[38] D. M. Powers, "Evaluation: From precision, recall and f-measure to roc, informedness, markedness and correlation", *Journal of Machine Learning Technologies*, vol. 1, no. 2, pp. 37–63, 2011.

[39] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition", *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989, ISSN: 0018-9219. DOI: 10.1109/5.18626.

[40] L. Rokach, "Ensemble-based classifiers", *Artificial Intelligence Review*, vol. 33, no. 1, pp. 1–39, 2010.

[41] S. M. Ross, *Introduction to probability models*, 10th ed. Amsterdam: Academic Press, 2010, ISBN: 978-0-12-375686-2.

[42] T. Rydén, T. Teräsvirta, and S. Åsbrink, "Stylized facts of daily return series and the hidden markov model", *Journal of applied econometrics*, pp. 217–244, 1998.

[43] ——, "Stylized facts of daily return series and the hidden markov model", *Journal of applied econometrics*, pp. 217–244, 1998.

[44] R. Sundberg, "Maximum likelihood theory for incomplete data from an exponential family", *Scandinavian journal of statistics : SJS ; theory and applications*, vol. 1, no. 2, pp. 49–58, 1974, ISSN: 03036898.

[45] P. Wang, "Markov zero-inflated poisson regression models for a time series of counts with excess zeros", *Journal of Applied Statistics*, vol. 28, no. 5, pp. 623–632, 2001.

[46] T. Weithers, *Foreign Exchange: A Practical Guide to the FX Markets*, 1st ed. New York: Wiley, 2006, ISBN: 978-0-47-173203-7.

[47] C. J. Wu, "On the convergence properties of the em algorithm", *The Annals of Statistics*, vol. 11, no. 1, pp. 95–103, 1983, ISSN: 00905364.

[48] C. Yuan, "Forecasting exchange rates: The multi-state markov-switching model with smoothing", *International Review of Economics & Finance*, vol. 20, no. 2, pp. 342–362, 2011.

[49] B. f. i. Zahlungsausgleich, "Triennial-central bank survey-report on global foreign exchange market activity in 2010", *The Bank for International Settlements*, 2010.

[50] Y. Zhang, "Prediction of financial time series with hidden markov models", PhD thesis, Applied Sciences: School of Computing Science, 2004.

[51] W. Zucchini and I. L. MacDonald, *Hidden Markov Models for Time Series: An Introduction Using R*, 2nd ed. London, UK: Chapman and Hall/CRC, 2016, ISBN: 978-1-48-225383-2.

TRITA TRITA-SCI-GRU 2018:005